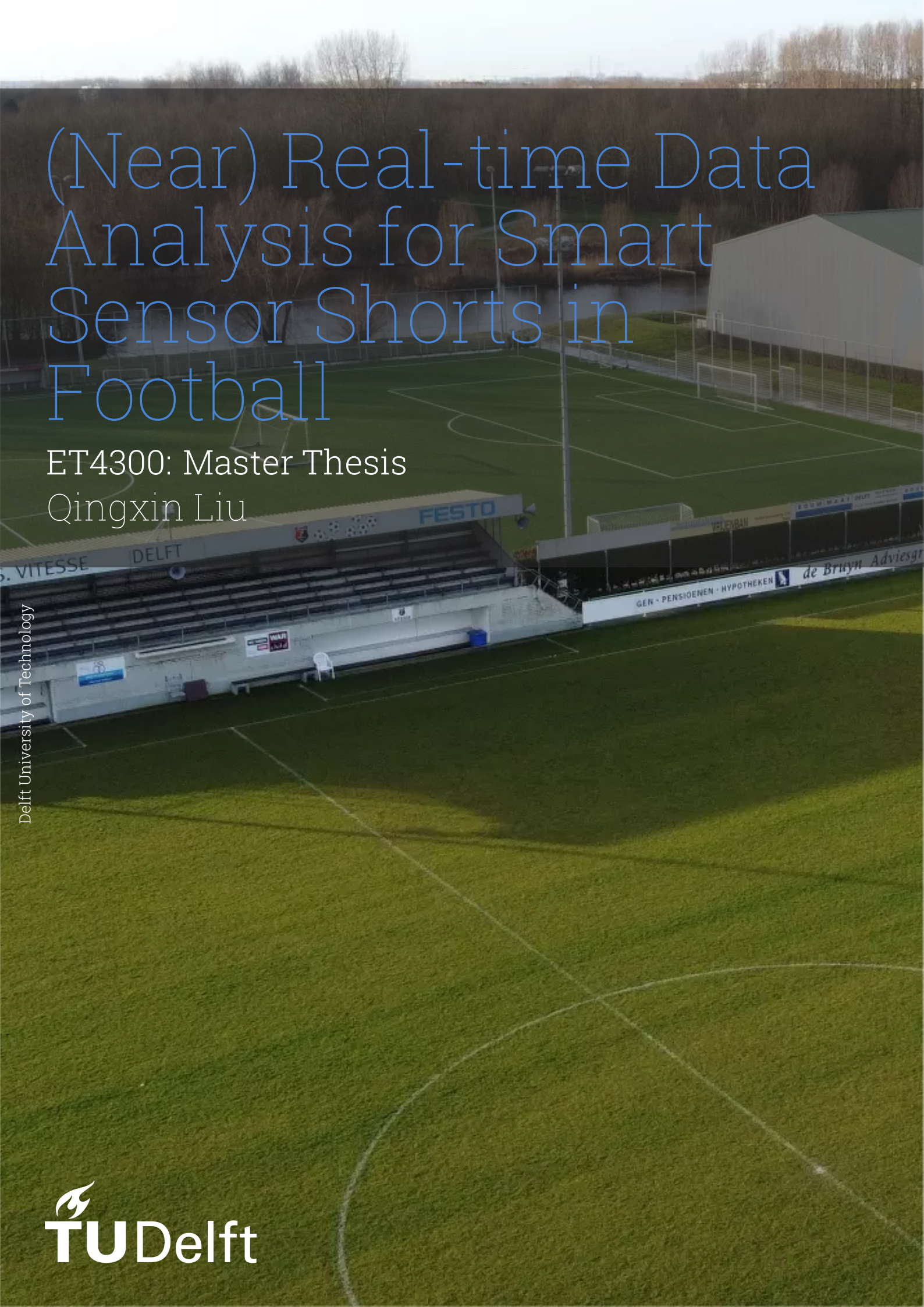


(Near) Real-time Data Analysis for Smart Sensor Shorts in Football

ET4300: Master Thesis

Qingxin Liu

Delft University of Technology

TU Delft

(Near) Real-time Data Analysis for Smart Sensor Shorts in Football

by

Qingxin Liu

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Friday July 21, 2023 at 10:00 AM.

Student number:	5549205	
Project duration:	November, 2022 – July, 2023	
Thesis committee:	Dr.ir. A. Bossche,	TU Delft, supervisor
	Prof. dr. ir. Jansen, K.M.B.,	TU Delft
	Prof. dr. P.J. French,	TU Delft

An electronic version of this thesis is available at http://repository.tudelft.nl/.

TU Delft

Preface

This thesis is written as part of my graduation project to obtain a master's degree in electrical engineering at Delft University of Technology. I am delighted and immensely grateful to have reached this significant milestone. Therefore, I would like to express my gratitude to the individuals who have supported me throughout this journey, leading to this achievement.

First and foremost, I would like to extend my heartfelt thanks to Andre Bossche, Jeroen Bastemeijer, Kaspar Jansen, and Annemarijn Steijlen for their guidance and mentorship throughout the project. Their expertise and patience have been instrumental in shaping the direction and success of this research.

I also want to thank Erik Wilmes and Bram Bastiaansen for their assistance and insightful explanations regarding the dataset used in this study. Their willingness to share their knowledge has greatly contributed to the quality of my research findings.

Furthermore, I would like to express my heartfelt appreciation to my parents for their unconditional love and belief in my abilities. Their constant encouragement and sacrifices have been the driving force behind my academic journey, and I am forever grateful for their support.

Last but certainly not least, I would like to extend my heartfelt thanks to my friend, Shirley, and my teammates at the DCF Football Club. Their unwavering support, passion, and encouragement have been instrumental in maintaining a healthy work-life balance during this project. Their presence has been a constant source of motivation and inspiration for me.

Qingxin Liu
Delft, July 2023

Abstract

Most injuries in football occur in the lower extremities due to high muscle stress. To prevent such injuries, the Dutch Football Association (KNVB) and the Delft University of Technology developed the Smart Sensor Shorts, an inertial sensor-based tracking system measuring the athlete's lower body kinematics, to improve physical load estimates during training sessions and matches. However, the system currently only has offline data analysis software, which results in poor monitoring capability.

This thesis proposes a near real-time data analysis system for Smart Sensor Shorts, featuring an automatic sensor calibration module, a football-specific activity recognition module, and a user interface, to monitor users' lower limb movement and load during football training. The proposed automatic sensor-to-body calibration algorithm maintains a high calibration accuracy with an $18.92°(\pm 5.74°)$ calibration error on average and simplifies the calibration process by leveraging detected standing and walking movements to estimate calibration parameters. The proposed gradient-boosting decision trees activity recognition model utilizes hip joint angles and joint angular velocities derived by the system to predict users' football-related activities, achieving an overall accuracy of $93.62\,\%$. The designed system processes the data recorded by IMUs in real time with a speed of 21 milliseconds per iteration and displays the calculated results related to the user's physical load on the user interface at a frame rate of 20 Hz.

Contents

1 Introduction

1.1. Background

Football players frequently face high physical demands during training and competition [1], [2], resulting in a relatively high incidence of injuries [3], [4]. To assist players, coaches, and medical staff in measuring physical load and reducing the risk of injuries, systems utilizing global navigation satellite systems (GNSS) and local positioning measurement systems (LPM) have been developed. These systems continuously track player positions on the field, allowing for the derivation of whole-body global displacements and the collection of metrics such as total distance covered and distance above specific speed thresholds.

However, the most prevalent type of injury in football is muscle injury in the lower extremities [5]. These injuries account for more than a third of all time-loss injuries in football and result in over a quarter of the total injury absences [6]. The high stress on lower limb muscles during explosive actions such as sprinting, cutting, jumping, and kicking in modern gameplay is a significant factor contributing to these injuries [7]. Since such actions do not involve significant on-field displacements, they cannot be captured by the aforementioned systems.

This challenge can be addressed by specifically measuring lower body kinematics during training and competition using the optoelectronic Motion Analysis System [8]. However, optoelectronic motion capture systems have certain limitations. Firstly, they require clear sight between the cameras and the markers. Secondly, these systems need a relatively large number of markers. Lastly, their use is confined to a limited measurement space. These limitations restrict the application of optoelectronic systems to laboratory settings and render them unsuitable for daily motion tracking during sports.

In order to tackle this challenge, a novel monitoring garment, the Smart Sensor Shorts, has been developed[9]. This garment incorporates three Inertial Measurement Units (IMUs) into sports tights to measure the movement of the pelvis and thighs. By integrating IMUs, the Smart Sensor Shorts eliminate the necessity for cameras and enable a direct on-field analysis of lower-body motion. The data utilized in this study was recorded and processed using a prototype of this inertial-based monitoring system [10]. The subsequent section will present the underlying principle of this system.

1.2. Inertial-based Motion Tracking System

1.2.1. System Setup

The inertial motion analysis system utilized in this study comprised three 9-DOF IMUs (MPU-9150, Invensense, San Jose, CA, USA) which were affixed to the pelvis and both thighs. The detailed specification of the sensor can be checked in Appendix A. Each IMU recorded triaxial acceleration, angular velocity, and magnetic field strength within the local coordinate frame attached to the sensor. Each sensor was encased in a protective housing along with a battery and an SD card, which logged the data at a sampling frequency of 500 Hz. The configuration of the system is shown in Figure 1.1.

Figure 1.1: Sensor placement [10]: The sensors were placed at the sacrum and iliotibial tracts.

1.2.2. Experiment Process

During the experiment, the participant conducted four football training drills [11], each lasting 4 minutes. The drill encompasses various activities, including standing, walking forward and backward, moving sideways, jogging forward and backward, cruising (sprinting at 80 % maximum intensity with direction changing), sprinting, heading, short-passing, and long-passing. To ensure adequate recovery between drills, a period of 3 minutes involving dribbling the ball was allotted for active rest. After the training, the participant proceeded to perform the calibration procedure [10], this procedure helps align the coordinate frame of each sensor with its corresponding body segment frame and involves sequentially raising the right leg, raising the left leg, and bowing forward.

1.2.3. Data Processing

The system uses the data recorded by the IMU sensors during the experiment to obtain the joint kinematics of the subject's left hip and right hip through the following steps:

1. Update the orientation quaternion of each sensor with respect to the global earth frame by using the Madgwick filter and data collected by IMU sensors.
2. Align the sensor coordinate frame to the corresponding body segment based on the calibration movements that were performed after the measurement period.
3. Obtain joint orientations by expressing the orientation of the distal body segment in the coordinate frame of the proximal segment for each joint.
4. Obtain joint angles by decomposing the obtained joint orientations into 'XZY' Euler angles.
5. Obtain the joint angular velocities by subtracting the angular velocity of the proximal segment from the gyroscope signal of the corresponding distal segment in the coordinate frame of the proximal segment.

1.3. Objective

Although the current system provided a simple method to obtain hip joint kinematics using IMUs and showed good validity with optoelectronic motion analysis system for football-specific movements, it still faces several challenges:

1. The calibration process of the system is intricate and time-consuming, making it difficult to implement for on-field monitoring.
2. The system is designed for offline analysis of data stored on the SD card, resulting in poor real-time capability.
3. The system solely calculates joint kinematics, which does not directly reflect the physical load experienced by the lower extremities of the players.

The primary objective of this study is to address the above issues. Firstly, an automatic algorithm for sensor-to-body calibration is developed, enabling unobtrusive calibration procedures. Additionally, a machine learning-based activity recognition model is devised to recognize highly intensive movements using hip joint kinematics, facilitating quantifying the player's physical load during football training. Lastly, a near real-time sensor data analysis system is designed where all algorithms operate frame by frame and the processed data is presented on a user interface, optimizing the overall monitoring experience.

The rest of this thesis is structured as follows: Chapter 2 will present an automatic sensor calibration method, detailing its methodology and implementation. Chapter 3 introduces a gradient-boosting tree model for human activity recognition in football, highlighting its construction and performance. Chapter 4 will comprehensively describe the pipeline of the proposed near real-time sensor data processing system. Furthermore, the design and utilization of the user interface will be extensively explained. The final two chapters will be dedicated to presenting and discussing the results of the implemented system, providing a thorough analysis and evaluation of its performance.

The code for sensor calibration is available in Appendix D and *https://github.com/qliu/sensor_calib*
The code for football activity recognition is available at *https://github.com/qliu/HAR*
The code for system integration is available at *https://github.com/qliu/System*
The code for the user interface is available at *https://github.com/qliu/UI*

2

Automatic Sensor Calibration

When attaching sensors to the body segments, their orientation to these segments is currently unknown. To ensure that the sensors can be positioned in any orientation on each body segment, it is necessary to align the coordinate frame of each sensor with its corresponding body segment frame through calibration procedures. The current sensor-to-body calibration method employed in the system consists of two consecutive steps. The first step involves five seconds of static standing. And the second step entails three functional calibration movements including raising the right upper leg, raising the left upper leg, and leaning the trunk forward. These procedures impose limitations on the practical application of the IMU-based system for on-field monitoring as they require an experienced operator to perform the aforementioned process and consume an unnecessary amount of time. Consequently, there is a need to introduce a simple yet effective sensor-to-body calibration procedure.

2.1. Related Works

To tackle the aforementioned issue, Nazarahari et al [12] proposed a method that relies solely on the IMU data collected during quiet standing and straight walking to perform sensor-to-body calibration, eliminating the need for calibration-specific movements. Although it still required prior knowledge of the standing and walking periods, this approach opens the possibility of a fully automatic calibration process as long as the standing and walking movements during football training and matches can be detected automatically.

Therefore, the current challenge lies in identifying and distinguishing the periods of standing and walking using the raw IMU signals. Luckily, recent studies [13], [14], [15] show that signal features calculated from the raw tri-axial accelerometer readouts perform well in classifying the intensity of physical activities. The next section will show the validity of using the mean amplitude deviation of the raw acceleration signal to determine the standing and walking movements during football training and introduce a fully automatic sensor calibration algorithm based on the above-mentioned standing and walking detection.

2.2. Method

2.2.1. Mean Amplitude Deviation

Mean amplitude deviation (MAD) is a time domain feature for signals and describes the mean distance of data points about the mean in an epoch. The MAD for a signal **r** is calculated as:

$$MAD = \frac{1}{n} \sum |r_i - \bar{r}| \tag{2.1}$$

where n is the number of samples in the epoch, r_i is the i^{th} signal sample within the epoch and $\bar{r}$ is the mean signal value of the epoch. In this study, **r** is chosen as the Euclidean norm of the pelvis sensor's accelerometer readouts:

$$r_i = \sqrt{a_{pel,x}^2 + a_{pel,y}^2 + a_{pel,z}^2} \tag{2.2}$$

where $a_{pel,x}, a_{pel,y}, a_{pel,z}$ represent the raw accelerometer readouts of the pelvis sensor along the x, y, and z-axis of the corresponding sensor reference frame. The MAD of the resultant norm signal is calculated within epochs with the size of 500 data points and 95% overlapping. The result is shown in Figure 2.1.

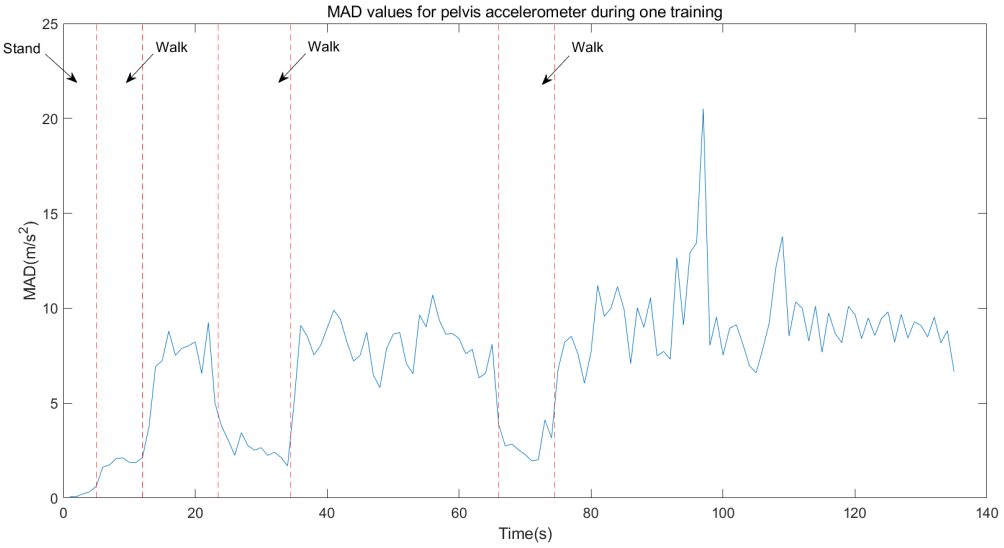


Figure 2.1: MAD value for accelerometer readouts during one football training. Epoch with 500 samples and 95% overlap.

The graph reveals that during low-intensity activities (standing and walking), the MAD value of raw accelerometer data is lower compared to other high-intensity activities, including jogging, jumping, kicking, and sprinting. Furthermore, there is a noticeable difference in the MAD value between standing and walking. Figure 2.2 presents the distribution of MAD values for various activity types during one training period, highlighting the effectiveness of MAD as a metric for identifying standing and walking from football-specific activities.

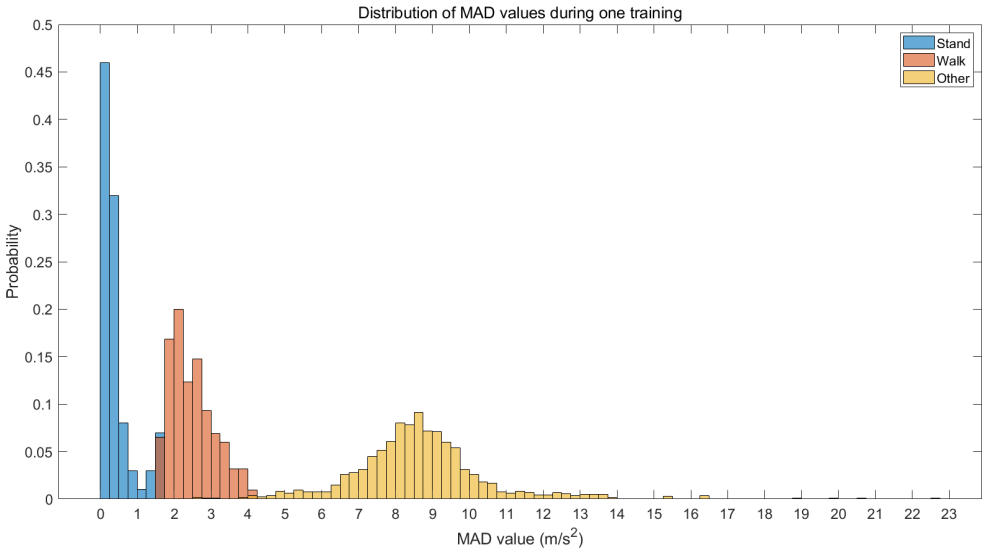


Figure 2.2: Distribution of MAD values for different movements. Standing (blue), walking (orange), and other high-intensity activities (yellow).

When choosing the MAD threshold to divide standing, walking, and other movements, the main issue that needs to be considered is the precision of each category instead of the recall, since the accuracy of

the sensor calibration depends on the accuracy with which the action is performed. Therefore, $[0, 1.5]$ is chosen as the interval to identify standing, and $[1.75, 3.5]$ is chosen as the interval to identify walking. The confusion matrix of the classification result on one football training using the abovementioned thresholds is shown in Figure 2.3. It can be seen that all predicted standing and walking movements are actual standing and walking.

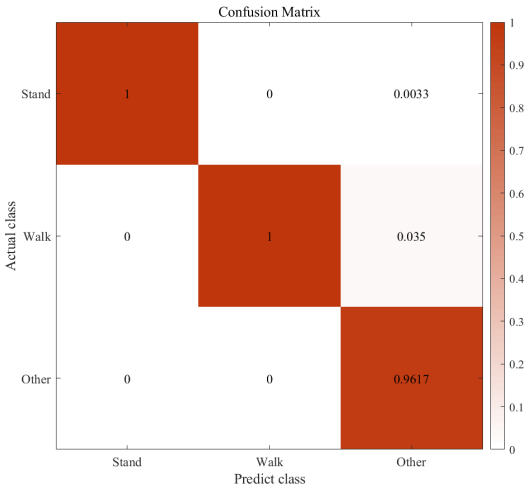


Figure 2.3: Confusion matrix of MAD-based standing/walking classification.

2.2.2. Sensor-to-body Calibration

After collecting the IMU's data during the standing and walking period, the next step is to align IMU's sensor frame (SF, shown in black in Fig 2.4a, also see Appendix A) with the corresponding body frame (BF, shown in green in Fig 2.4a) as recommended by the International Society of Biomechanics (ISB) [16]. This thesis follows the same procedures as [12] suggests: quiet standing for three to five seconds (Vertical Calibration) followed by eight steps of level walking in a straight line (Heading Calibration). Two procedures are shown in Figure 2.4b and described below.

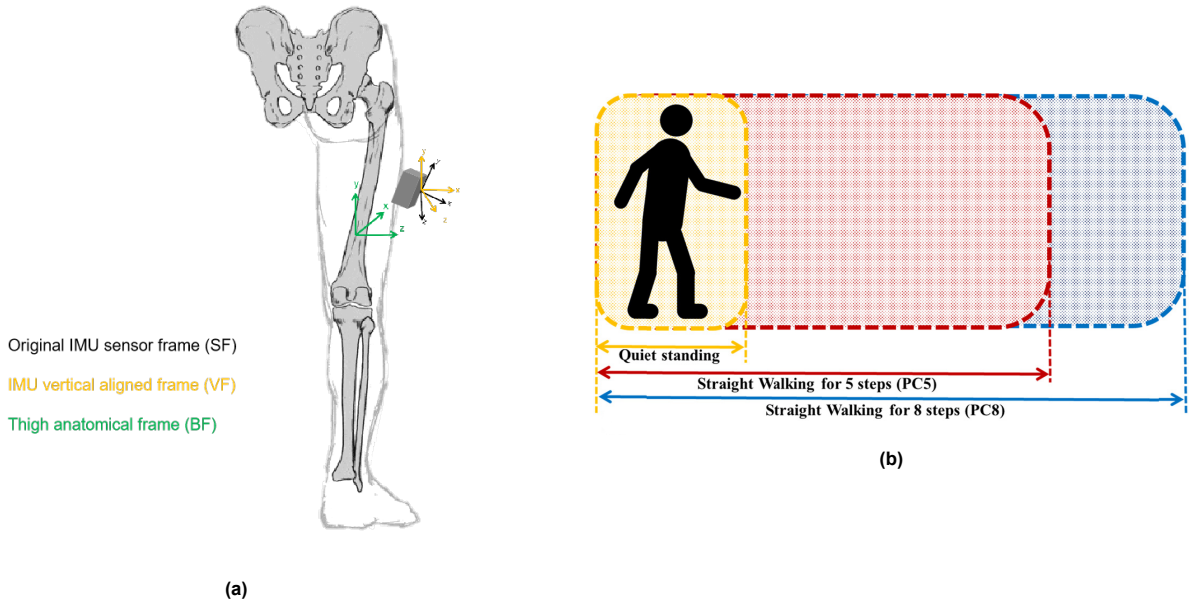


Figure 2.4: 2.4a. Different reference frames using in sensor calibration (rear view). 2.4b. Calibration procedure [12]: Vertical Calibration followed by Heading Calibration.

Vertical Calibration

Vertical Calibration is performed first. During this period, the transformation ${}^{VF}_{SF}R$ is found to align IMU's vertical axis with the gravitational acceleration. ${}^{VF}_{SF}R$ represents the rotation matrix of the sensor frame (SF) with respect to the vertically aligned frame (VF, shown in yellow in Fig 2.4a). The VF for each sensor is defined as follows: the y-axis is pointing upward, parallel to the direction of gravity; the x-axis is pointing forward, parallel to the ground and perpendicular to the z-axis of the SF, and the new z-axis completes a right-handed Cartesian coordinate system.

${}^{VF}_{SF}R$ is composed of rotations around the x-axis and z-axis of the original IMU's SF. Assuming during quiet standing only gravitational acceleration is measured by each sensor, the mean acceleration along the x-axis and z-axis of the VF are both equal to zero, and the mean acceleration along the y-axis of the VF is equal to gravitational acceleration, ${}^{VF}_{SF}R$ can be obtained as:

$$ {}^{VF}_{SF}R = \begin{bmatrix} \frac{a_y^2+a_z^2}{a_x^2+a_y^2+a_z^2} & \frac{-a_x a_y}{a_x^2+a_y^2+a_z^2} & \frac{-a_x a_z}{a_x^2+a_y^2+a_z^2} \\ \frac{a_x}{\sqrt{a_x^2+a_y^2+a_z^2}} & \frac{a_y}{\sqrt{a_x^2+a_y^2+a_z^2}} & \frac{a_z}{\sqrt{a_x^2+a_y^2+a_z^2}} \\ 0 & \frac{-a_z}{\sqrt{a_x^2+a_y^2+a_z^2}} & \frac{a_y}{\sqrt{a_x^2+a_y^2+a_z^2}} \end{bmatrix} \tag{2.3} $$

where (a_x, a_y, a_z) is the mean value of the accelerometer readouts during the quiet standing period. It should be noticed here that for each sensor, the x-axis of its VF is not like the x-axis of its corresponding body frame (BF) which is always perpendicular to the frontal plane of the human body. Therefore, the heading calibration is performed next to fix the above orientation error between the two frames.

Heading Calibration

During the heading calibration, the vertically aligned frame (VF) of each IMU is aligned with the corresponding body frame (BF). This is done by finding a single-axis rotation ${}^{BF}_{VF}R$ around the y-axis of VF. Again, ${}^{BF}_{VF}R$ represents the rotation matrix of the VF with respect to the BF. Assuming that walking along a straight line replicates a single-axis rotation (around the z-axis of the corresponding BF) for two thigh IMUs, the direction of the z-axis of BF, which maximized the variations of angular velocity, can be found by applying Principal Component Analysis (PCA) to the angular velocity recorded by each IMU during straight walking in its VF's x-z plane.

To this end, for two thigh sensors, the rotation angle around the y-axis of the corresponding VF can be written as equation 2.4, and the transformation can be further expressed as equation 2.5.

$$ \gamma = tan^{-1}(\frac{C(2,1)}{C(1,1)}) \tag{2.4} $$

$$ {}^{BF}_{VF}R = \begin{bmatrix} cos\,\gamma & 0 & sin\,\gamma \\ 0 & 1 & 0 \\ -sin\,\gamma & 0 & cos\,\gamma \end{bmatrix} \tag{2.5} $$

where C is the principal component coefficient matrix of the measured angular velocity in VF's x-z plane. Since the gyroscope readouts were not directly used to estimate the orientation, gyroscope drift will not affect the proposed calibration. With the above calibration procedures, the final calibration matrix for two thigh IMUs can be calculated by multiplying the two previously obtained transformations together as equation 2.6.

$$ {}^{BF}_{SF}R = {}^{BF}_{VF}R \cdot {}^{VF}_{SF}R \tag{2.6} $$

However for the pelvis sensor, there is no single-axis rotation movement for the pelvis in real football training; therefore, its vertically aligned frame (VF) is not aligned with its body frame (BF) during walking, but using three IMUs' magnetometer readouts during quiet standing instead.

Specifically, assuming the horizontal components of the earth's magnetic field at three sensor locations are the same, since three body frames have the same orientation during the standing period, the magnetic field vectors measured by three sensors should be identical in the x-z plane of the corresponding

body frames as well. With two already calibrated thigh IMUs, the angle ϕ between the horizontal earth magnetic vector and the z-axis of the body frame can be calculated as:

$$\phi = \frac{tan^{-1}(\frac{^{BF}m_{x,left}}{^{BF}m_{z,left}}) + tan^{-1}(\frac{^{BF}m_{x,right}}{^{BF}m_{z,right}})}{2} \tag{2.7}$$

where $^{BF}m_{x,left}$,$^{BF}m_{z,left}$ and $^{BF}m_{x,right}$,$^{BF}m_{z,right}$ represent the mean magnetic field strength along the x-axis and z-axis of the body frame during the quiet standing period measured by left and right thigh sensors respectively.

The direction of the horizontal earth magnetic vector with respect to the pelvis VF's z-axis ϕ_{pelvis} can be calculated following the same procedure. And the final rotation angle γ_{pelvis} for the pelvis senor around the y-axis of its VF is derived by taking the difference between the two angles obtained above:

$$\phi_{pelvis} = tan^{-1}(\frac{^{VF}m_{x,pelvis}}{^{VF}m_{z,pelvis}}) \tag{2.8}$$

$$\gamma_{pelvis} = \phi_{pelvis} - \phi \tag{2.9}$$

where $^{VF}m_{x,pelvis}$ and $^{VF}m_{z,pelvis}$ represent the mean magnetic field strength along the x-axis and z-axis of pelvis vertically aligned frame during the standing period. With the γ_{pelvis}, the transformation matrix $^{BF}_{VF}R$ for the pelvis sensor can be derived using equation 2.5 as well.

2.2.3. Joint Kinematics

After the calibration, the angle and angular velocity of two hip joints can be calculated. The whole process is shown in Figure 2.5. The quaternion $^{SF}_{GF}q$ representing the orientation of each sensor with respect to the global reference frame (GF) derived by the Madgwick filter [17] is first aligned to the corresponding body frame (BF):

$$^{BF}_{GF}q = ^{SF}_{GF}\ q \otimes ^{BF}_{SF}\ q_{calib} \tag{2.10}$$

where $^{BF}_{SF}q_{calib}$ is the calibration quaternion converted from the calibration matrix $^{BF}_{VF}R$ derived in the previous section, and the global reference frame is defined as follows: The y-axis is pointing upward, parallel to the direction of gravity; the x-axis is pointing in the direction of the horizontal component of the earth's magnetic field vector; and the z-axis completes a right-handed Cartesian coordinate system.

Then the orientation of each joint (left or right hip) can be extracted by calculating the relative orientation between its distal segment (ipsilateral thigh) and its proximal segment (pelvis):

$$^{BF-PB}_{BF-DB}q_{joint} = ^{BF-DB}_{GF}\ q^{-1} \otimes ^{BF-PB}_{GF}\ q \tag{2.11}$$

where $^{BF-DB}_{GF}q$ denotes the orientation of the distal segment with respect to the global reference frame, and $^{BF-PB}_{GF}q$ denotes the orientation of the proximal segment with respect to the global reference frame.

Finally with the obtained joint orientation quaternion, 3D hip joint angles can be extracted by simply converting the quaternion to Euler angles. And hip joint angular velocities can be obtained by subtracting the angular velocity of the proximal segment (pelvis) from the angular velocity of the distal segment (ipsilateral thigh):

$$\omega^{BF-PB}_{joint} = ^{BF-PB}_{BF-DB}\ q_{joint} \otimes \omega^{BF-DB} \otimes ^{BF-PB}_{BF-DB}\ q^{*}_{joint} - \omega^{BF-PB} \tag{2.12}$$

where ω^{BF-DB} and ω^{BF-PB} denote the measured angular velocity of the proximal segment and the distal segment, respectively, while $^{BF-PB}_{BF-DB}q_{joint} \otimes \omega^{BF-DB} \otimes ^{BF-PB}_{BF-DB}\ q^{*}_{joint}$ transforms the angular velocity of the distal segment into the coordinate frame of the proximal segment. It is noteworthy that the hip joint coordinate system adheres to the ISB recommendation [18] and has the same initial orientation as the pelvic coordinate system.

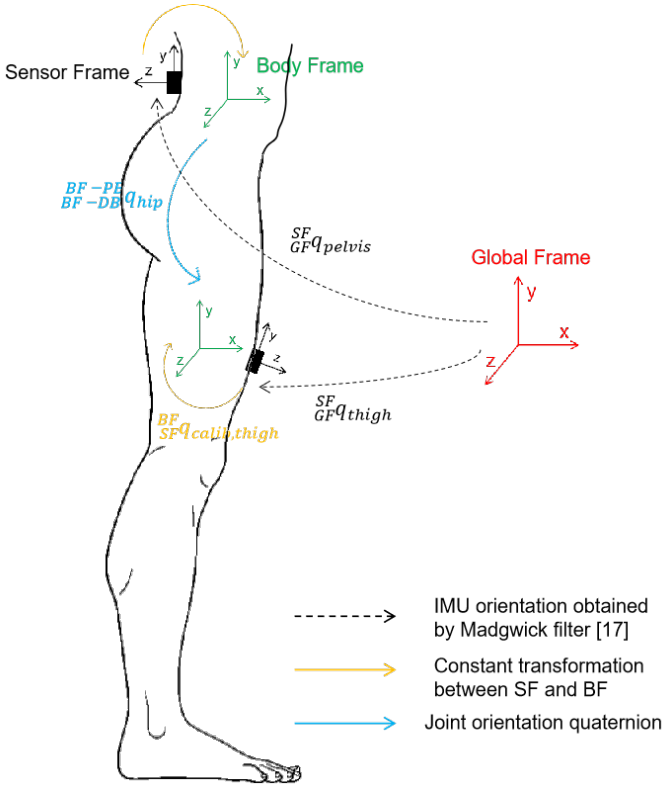


Figure 2.5: Joint kinematics calculation: (i). obtain IMU orientation quaternion ${}^{SF}_{GF}q$ (black); (ii). calculate the orientation of the corresponding body segment ${}^{BF}_{GF}q$ (yellow); (iii). derive the quaternion of the joint in between ${}^{BF-PB}_{BF-DB}q_{joint}$ (blue); (iv). compute joint angles and angular velocities based on joint orientation.

2.3. Result[1]

To validate the feasibility of the proposed sensor calibration method and evaluate the influence of the two calibration stages on the final result, four sets of calibration parameters are computed following the two-stage calibration procedure. The first set is the baseline and was derived using the calibration-specific movements [10]. The second set explores the effect of vertical calibration and was derived using the standing IMU data in [10] and walking IMU data detected from football training. The third set explores the effect of heading calibration, which was derived using the leg raising IMU data in [10] and standing IMU data detected from football training. The final set was derived fully automatically according to the proposed pipeline, using the IMU data corresponding to the standing and walking period identified from football training based on the raw pelvis accelerometer signal.

The orientation difference between the estimated calibration matrix and the reference calibration matrix (obtained by Wilmes et al [10]) is quantified using helical angles [19] as equation 2.13.

$$\theta = cos^{-1}\left(\frac{trace\left(\left({}^{BF}_{SF}R_0\right)^T \cdot \left({}^{BF}_{SF}R_1\right)\right) - 1}{2}\right) \tag{2.13}$$

where ${}^{BF}_{SF}R_1$ is the estimated calibration matrix and ${}^{BF}_{SF}R_0$ is the reference calibration matrix. The value of θ ranges in $[0, \pi]$. The smaller the θ, the closer ${}^{BF}_{SF}R_1$ is to ${}^{BF}_{SF}R_0$, and the better the calibration is.

The result at each sensor location for Subject 3 is given in Table 2.1 and Figure 2.6. Since the subject engaged in multiple sets of training, the second to the fourth set parameters are estimated three times using standing and walking periods identified from the first, second, and fourth training rounds, and

[1]The experimental results in this section are derived using the data recorded for Subject 3 in the provided dataset. The detailed specification of the used data can be found in appendix B.

the final results are given in the 'mean(±std)' form. The results for the other two subjects are given in appendix C.

Table 2.1: Helical angles (°) [19] between obtained calibration matrix and reference.

	Pelvis	Left Thigh	Right Thigh
Stand[10]+Raise Leg[10]	18.76	2.38	1.02
Stand[10]+Walk	14.02(±1.49)	7.73(±1.73)	15.85(±1.29)
Stand+Raise Leg[10]	22.75(±7.29)	10.02(±0.99)	15.54(±6.46)
Stand+walk	17.57(±8.20)	12.10(±1.37)	22.85(±3.91)

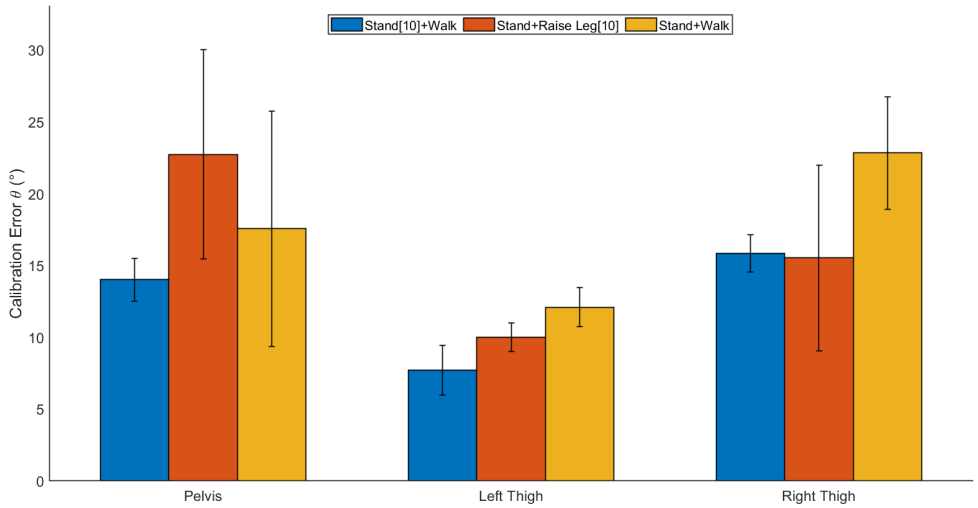


Figure 2.6: Bar plot of mean and standard deviation values of the calibration error for subject 3 on three sensor locations.

Furthermore, to explore the effect of sensor calibration errors on joint kinematics calculation, the difference in hip joint angle and joint angular velocity along each axis, resulting from the proposed calibration method and the reference method [10], is examined for Subject 3 under different activity types across four training rounds (see Appendix B) and quantified as the root mean square error (RMSE):

$$RMSE = \sqrt{\frac{1}{T-1}\sum_{i=1}^{T}\left(J_{0,i} - J_{1,i}\right)^2} \tag{2.14}$$

where T is the duration of the calculation interval, and J_0 and J_1 are the reference and estimated joint kinematics time-series, respectively. The results are shown in Table 2.2 and 2.3 and given in the 'mean(±std)' form.

A five-second example of hip joint angles derived without calibration, with the proposed calibration method, and with the reference calibration method [10] during one training is shown in Figure 2.7. This figure suggests the proposed calibration enables the Smart Sensor Short system to track the user's joint kinematics with good accuracy.

Table 2.2: Root mean square error (RMSE) of left/right hip angles $\theta(^\circ)$.

Movement	$\theta_{left,x}$	$\theta_{left,y}$	$\theta_{left,z}$	$\theta_{right,x}$	$\theta_{right,y}$	$\theta_{right,z}$
Walk	$12.85(\pm0.47)$	$4.56(\pm0.08)$	$6.00(\pm0.54)$	$7.76(\pm0.48)$	$22.13(\pm0.30)$	$5.30(\pm0.17)$
Jog	$12.25(\pm0.24)$	$6.27(\pm0.26)$	$6.30(\pm0.57)$	$8.49(\pm0.31)$	$21.83(\pm0.10)$	$5.77(\pm0.18)$
Jump	$12.68(\pm0.21)$	$8.02(\pm0.07)$	$8.43(\pm0.69)$	$9.34(\pm0.24)$	$21.20(\pm0.57)$	$8.62(\pm0.70)$
Kick	$13.02(\pm0.27)$	$8.35(\pm1.00)$	$7.89(\pm1.10)$	$10.59(\pm0.72)$	$22.24(\pm0.48)$	$10.20(\pm1.23)$
Sprint	$12.81(\pm0.13)$	$8.38(\pm0.42)$	$11.26(\pm1.31)$	$10.02(\pm0.36)$	$21.09(\pm0.18)$	$13.34(\pm1.21)$
Overall	$12.54(\pm0.22)$	$6.46(\pm0.20)$	$7.21(\pm0.53)$	$8.70(\pm0.28)$	$21.81(\pm0.16)$	$7.47(\pm0.45)$

Table 2.3: Root mean square error (RMSE) of left/right hip angular velocities $\omega(^\circ/s)$.

Movement	$\omega_{left,x}$	$\omega_{left,y}$	$\omega_{left,z}$	$\omega_{right,x}$	$\omega_{right,y}$	$\omega_{right,z}$
Walk	$68.12(\pm4.18)$	$141.70(\pm5.68)$	$75.59(\pm3.34)$	$85.48(\pm3.96)$	$141.41(\pm9.21)$	$81.61(\pm3.05)$
Jog	$123.86(\pm5.61)$	$203.20(\pm11.60)$	$110.21(\pm3.06)$	$111.65(\pm3.74)$	$172.01(\pm4.48)$	$127.79(\pm2.78)$
Jump	$166.40(\pm6.80)$	$345.02(\pm22.55)$	$183.36(\pm11.30)$	$168.56(\pm18.22)$	$283.50(\pm28.09)$	$198.86(\pm17.57)$
Kick	$234.01(\pm36.23)$	$319.86(\pm42.17)$	$162.00(\pm24.97)$	$222.70(\pm27.12)$	$386.79(\pm46.31)$	$225.60(\pm25.90)$
Sprint	$264.18(\pm13.52)$	$369.81(\pm18.15)$	$238.40(\pm14.58)$	$233.72(\pm17.84)$	$331.28(\pm18.25)$	$260.34(\pm26.63)$
Overall	$148.61(\pm9.18)$	$235.70(\pm13.04)$	$134.80(\pm7.84)$	$140.53(\pm8.75)$	$221.12(\pm10.14)$	$154.26(\pm9.35)$

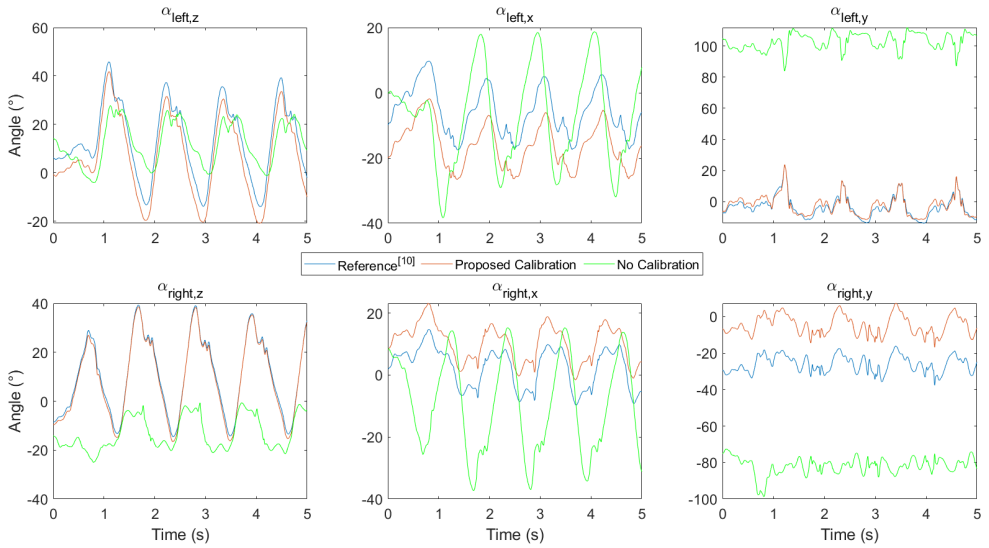


Figure 2.7: 5 seconds estimated (with and without calibration) and reference [10] hip angles during one training.

To see the effect of vertical alignment, the triaxial accelerations of three sensors during the quiet standing period are shown in Figure 2.8. After the calibration, for each sensor, the acceleration along the y-axis of the corresponding body frame is equal to the gravitational acceleration, and the accelerations along the other two axes are around zero. To see the effect of heading alignment, the triaxial angular velocities of the left and right thigh sensors during the walking period are shown in Figure 2.9. After the calibration, the variance of the angular velocities along the corresponding body frame's z-axis is maximized for both sensors.

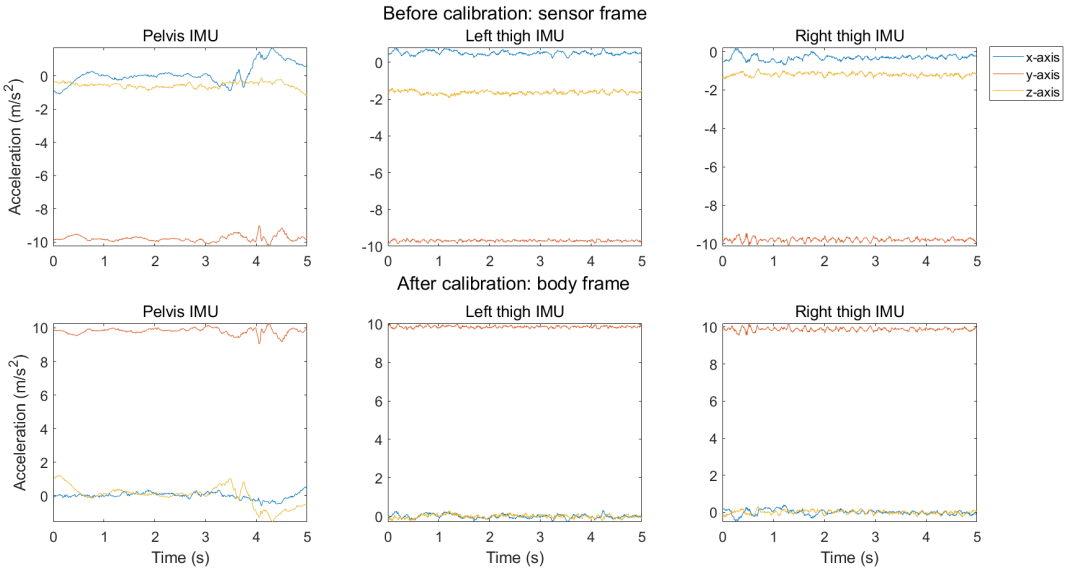


Figure 2.8: Accelerations during quiet standing in the sensor frame (before calibration) and body frame (after calibration).

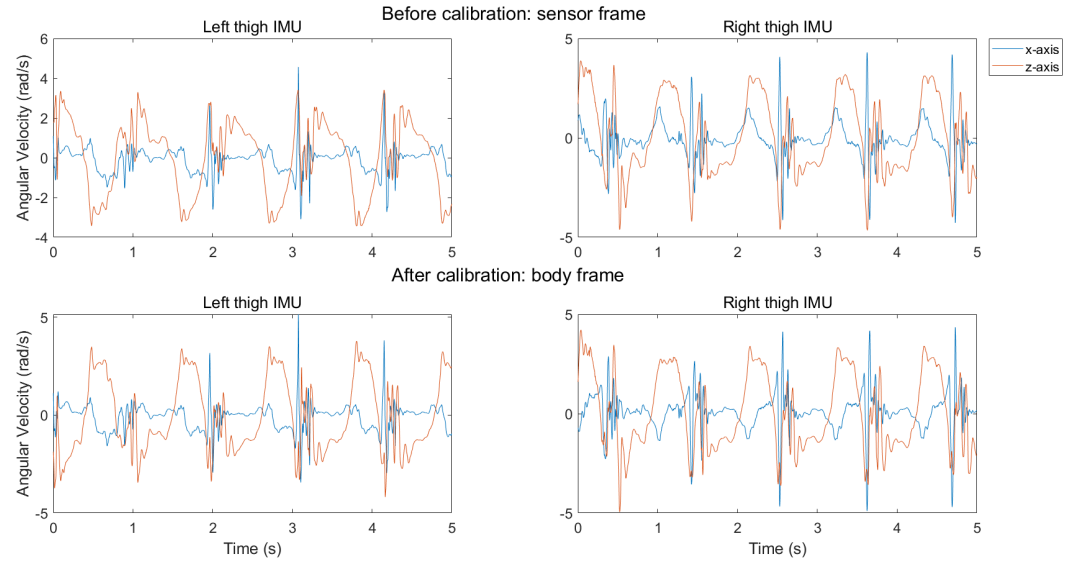


Figure 2.9: Angular velocities during walking in the sensor frame (before calibration) and body frame (after calibration).

2.4. Discussion

It can be seen from the table 2.1 that when using the calibration-specific movement, the estimated calibration parameters for two thigh sensors have good accordance with the reference parameters, proving the effectiveness of the two-stage calibration method. The pelvis sensor has a larger calibration error compared to the other two sensors since the second step of its calibration procedure depends on the magnetometer readouts and will be affected by interference from external magnetic fields. When partially or fully using the automatically extracted movements, the calibration accuracy of all three sensors is affected to varying degrees, but still acceptable. This performance drop is assumed due to the following reasons: First, the subject didn't stay strictly in a neutral upright pose when standing during the training, which can affect the precision of the vertical calibration; Second, compared to the leg raising in calibration-specific movement, it is more difficult to extract the principal component of the angular velocity from the walking movement since the subject swung both legs quicker and in a more subtle way, and hitting the ground while walking introduces vibrations in muscles and therefore produces

fluctuations in The the gyroscope readouts along other axes due to soft tissue artifact; Third, the main assumption, i.e., single-axis rotation of thigh in the sagittal plane during straight walking was not always satisfied since subjects have different gaits, which can affect the accuracy of the heading calibration. It can also be observed from Figure 2.6 that the accuracy of the vertical calibration has more impact on the final calibration result compared to the heading calibration, suggesting the necessity of a more strict still-standing identification from the raw accelerometer data.

Meanwhile, a performance difference is observed between the right thigh and left thigh sensor for Subject 3; however, no such difference is observed on the other two subjects. The reason for this could be that the right thigh sensor was not firmly attached to subject 3's body or subject 3 has an out-toeing in the right leg due to restricted internal rotation of the right hip. Both situations can cause non-single-axis rotation of the thigh during walking, and subsequently bring a performance degradation in right thigh sensor calibration. Also, it can be noticed that the calibration accuracy of the pelvis sensor is close to the worse one of the two thigh sensors as the second calibration stage of the pelvis sensor depends on the other two sensors' calibration result.

In Table 2.2, a significant effect of the movement type on the norm of three-axis hip joint angles RMSE is found for both left ($p < 0.001, \eta^2 = 0.872$) and right ($p < 0.001, \eta^2 = 0.867$) legs using One-way analysis of variance (ANOVA) method. And in Table 2.3, a significant effect of the movement type on the norm of three-axis hip joint angular velocities RMSE is found for both left ($p < 0.001, \eta^2 = 0.960$) and right ($p < 0.001, \eta^2 = 0.949$) legs. A larger deviation between the calculated joint kinematics and the reference is observed as well when the subject conducted physically demanding activities (i.e., kicking, jumping, and sprinting); therefore it can be concluded that the joint kinematics error caused by calibration error has a positive correlation with the intensity of the movement.

3

Football Activity Recognition

Obtaining information about the activities performed by players during a match or training can greatly enhance the analysis of their performance for coaches and medical staff. A comprehensive evaluation of the player's movements and exertion levels offers teams the opportunity to improve training planning, provide personalized monitoring for each player, and potentially prevent and treat injuries. Traditionally, the classification of activities relied heavily on camera assistance, which necessitated a complex and costly setup, limiting its accessibility to elite teams. Moreover, using cameras posed challenges when players were positioned behind one another, rendering it difficult to identify the specific player.

Fortunately, the development of the Smart Sensor Shorts presents a promising way for a cost-effective activity recognition system that can be within the reach of smaller teams. This chapter explores the utilization of a machine learning-based model for recognizing football activities, leveraging the hip joint kinematics (joint angles and joint angular velocities) obtained from the inertial-based motion tracking system introduced earlier.

3.1. Related Works

Identifying and recognizing human activities using signals acquired from IMU is a subject of interest in the fields of machine learning and signal processing, as evidenced by recent research efforts. In the beginning, conventional machine learning approaches such as Decision Trees, Support Vector Machines, and K-Nearest Neighbors were employed to classify activities [20], [21], [22]. These methods involved a manual selection or extraction of signal features, often derived from the time and frequency domains. Metrics such as mean, standard deviation, and coefficients of the Fast Fourier Transform were frequently utilized. However, computing capabilities and algorithms advancements have paved the way for a shift toward Deep Learning methodologies in this domain. Deep architectures, including convolutional Neural Networks (CNN) and Long Short-Term Memory Networks (LSTM), have gained popularity due to their ability to handle time series data effectively [23], [24], [25].

Although Deep Learning algorithms generally outperform traditional machine learning approaches and can automatically extract relevant features from raw IMU signals, there are specific disadvantages of the former in the context of the inertial-based tracking system investigated in this thesis:

1. The implementation of deep learning methods has primarily focused on daily activities and has not been extensively applied to football-related activities. Given the limited dataset available, there are insufficient samples to train a robust deep model.
2. Deep learning models tend to have more parameters, making them challenging to deploy on resource-constrained devices such as microcontrollers. Additionally, their higher computational demands can be power-consuming, which is undesirable in certain applications.
3. The evaluation process of deep learning methods is time-consuming, making them less suitable for the near-real-time application studied in this thesis.

4. The inertial-based tracking system utilized in this study can derive joint kinematics from raw IMUS signals. As a result, it incorporates built-in feature extraction, eliminating the need for additional feature selection or extraction.
5. Although the deep learning models can extract signal features automatically, these features are often high-level and less explainable.

Considering these factors, the traditional machine learning approach holds certain advantages within the scope of the near-real-time inertial-based tracking system. However, an ensemble method needs to be employed to improve the current model performance to match that of deep learning models. Therefore, this thesis mainly focuses on developing an ensemble model using the Classification And Regression Tree (CART) as the basic classifier and gradient boosting as the ensemble technique for human activity recognition in football-specific training.

3.2. Method

3.2.1. Gradient Boosting Decision Trees

Prior to the emergence and widespread adoption of deep learning, Gradient Boosting Decision Trees (GBDT) were regarded as a highly effective machine learning model that found extensive application across various domains. GBDT uses Classification and Regression Trees (CART) [26] as its base classifier. The CART algorithm selects one input feature at a time and identifies a threshold to bifurcate the samples into two groups. The same feature can be reused within the CART tree.

GBDT ensembles base classifiers through the gradient boosting method [27]. Unlike other boosting techniques that assign varying weights to weak classifiers based on their classification accuracy or adjust the data weights to reinforce misclassified samples (e.g., Adaboost), the gradient boosting functions as an additive model. Each newly grown tree learns the residual between the previous model output and the desired output. The final prediction of the complete model is obtained by summing the prediction results of all CART trees. The diagram of the GBDT classification model on one class is shown in Figure 3.1. The model training and test phases will be illustrated in the following sections.

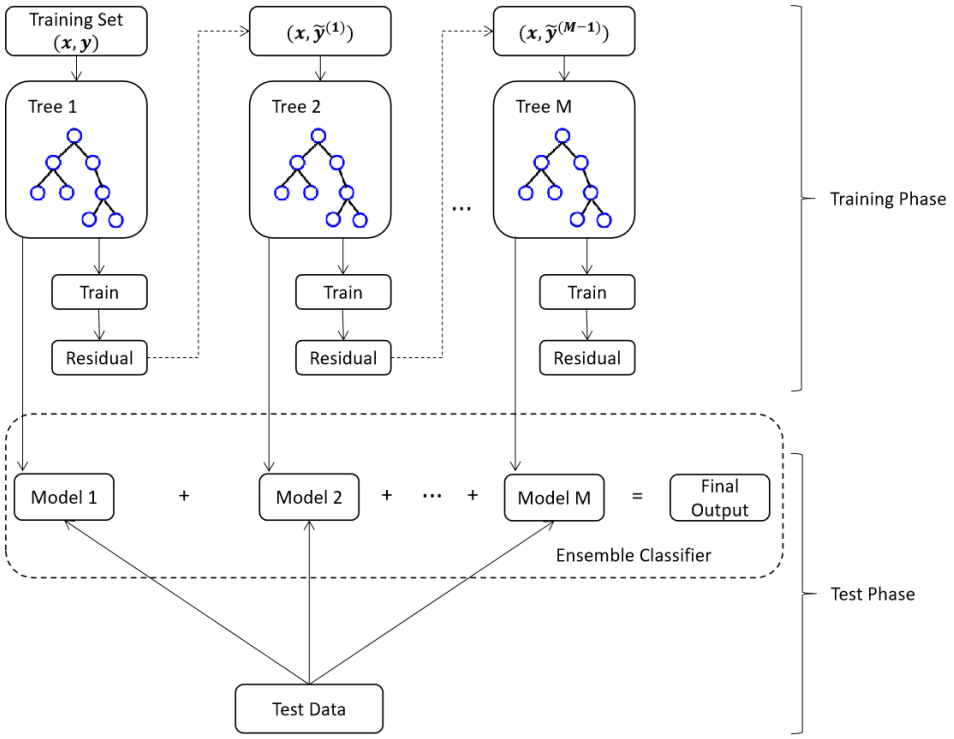


Figure 3.1: Diagram of Gradient Boosting Decision Trees [28].

3.2.2. Model Training

Data Preparation

Since only Subject 3 has the video recording, the joint kinematics of Subject 3 during the training period are calculated for model training and testing, using the calibration parameters obtained in the previous chapter. The derived joint data is manually labeled with the corresponding movement type according to the video. The activities performed during the football drill are divided into five different categories as Table 3.1 shows.

To generate training data for the model, a sliding window with a size of 500 samples (1 second) and 90% overlapping (100 ms) is used. The mean values of six joint kinematics in i^{th} window (3-axis joint angles and 3-axis joint angular velocities for left and right hip) are set as the feature $\mathbf{x}_i$ of data point i, and the majority class in the corresponding window is set as the label y_i of sample i. Also, considering the imbalanced class (the subject jogged and walked most of the time), a stratified sampling strategy is applied by selecting the first three training sessions for model training and the last session for model testing (see Appendix B) as the subject conducted the same set of activities during each session.

Table 3.1: Definition of different activity types.

Class Number	Class Name	Activity(s) Included
1	Low	Stand, Walk Forward, Walk Backward
2	Jog	Sideways, Jog Forward, Jog Backward
3	Jump	Heading
4	Kick	Short Pass, Long Pass
5	Sprint	Cruise, Sprint

Gradient Tree Boosting Algorithm

For the K-class classification GBDT, the goal in the model training phase is to build M regression trees for each class to predict the corresponding class probability for each input sample. Figure 3.2 shows the process of growing M trees for one specific class.

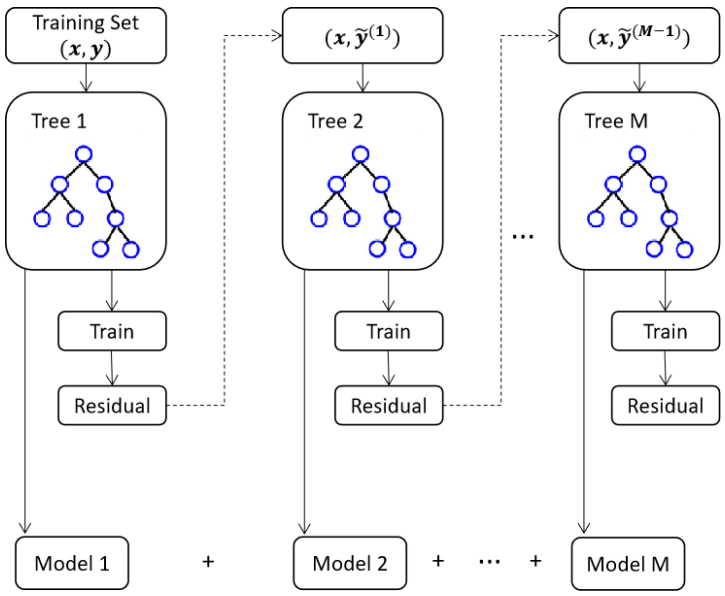


Figure 3.2: Training phase of Gradient Boosting Decision Trees.

The algorithm of K-class logistic gradient boosting is shown below.

Algorithm 1 Gradient Tree Boosting Algorithm

Initialize: $F_{k0}(\mathbf{x}) = 0, k = 1, 2, ..., K$.
for $m = 1$ *to* M **do**
  $p_k(\mathbf{x}) = \frac{\exp(F_k(\mathbf{x}))}{\sum_{l=1}^{K} \exp(F_l(\mathbf{x}))}, k = 1, 2, ..., K$; ▷ Predict class k 's probability using current model
  for $k = 1$ *to* K **do**
    $\tilde{y}_{ik} = y_{ik} - p_k(\mathbf{x}_i), i = 1, 2, ..., N$; ▷ Calculate the model residual for class k
    $\{R_{jkm}\}_{j=1}^{J} = J$-terminal node tree $\left(\{\tilde{y}_{ik}, \mathbf{x}_i\}_1^N\right)$; ▷ Build a tree for class k predicting the residual
    $\gamma_{jkm} = \frac{K-1}{K} \frac{\sum_{\mathbf{x}_i \in R_{jkm}} \tilde{y}_{ik}}{\sum_{\mathbf{x}_i \in R_{jkm}} |\tilde{y}_{ik}|(1-|\tilde{y}_{ik}|)}, j = 1, 2...J$; ▷ Find the optimal parameters for the newly built tree
    $F_{km}(\mathbf{x}) = F_{k,m-1}(\mathbf{x}) + \sum_{j=1}^{J} \gamma_{jkm} \mathbf{1}(\mathbf{x} \in R_{jkm})$; ▷ Update the model by adding the newly built tree
  end for
end for

K:	Number of classes	M:	Number of trees for each class
$p_k(\mathbf{x})$:	Predicted probability for class k	$F_k(\mathbf{x})$:	Current model's output on class k
y_{ik}:	True probability of sample i belonging to class k	N:	Number of samples
$\tilde{y}_{ik}$:	Negative gradient of the loss function	R_{jkm}:	Terminal region of the m^{th} tree for class k
γ_{jkm}:	Output of the j^{th} leaf node on the m^{th} tree for class k	F_{km}:	Model derived after m iterations for class k

After initialization, one tree is built for each class during each iteration to fit the residual of the current model. The model residual is derived by the loss function and can be computed as:

$$L\left(\{y_k, F_k(\mathbf{x})\}_1^K\right) = -\sum_{k=1}^{K} y_k \log p_k(\mathbf{x}) \tag{3.1}$$

where $y_k = 1(class = k) \in \{0, 1\}$, $F_k(\mathbf{x})$ represents the model output class probability for class k, and $p_k(\mathbf{x}) = Pr(y_k = 1 \mid \mathbf{x})$. Using the softmax function, the class probability $p_k(\mathbf{x})$ can be written as :

$$p_k(\mathbf{x}) = \exp(F_h(\mathbf{x})) / \sum_{l=1}^{K} \exp(F_l(\mathbf{x})) \tag{3.2}$$

The gradient boosting algorithm uses the 'line search' strategy to approach its training target. The negative gradient of the loss function defines the search direction and has the form :

$$\tilde{y}_{ik} = -\left[\frac{\partial L\left(\{y_{il}, F_l(\mathbf{x}_i)\}_{l=1}^K\right)}{\partial F_k(\mathbf{x}_i)}\right]_{\{F_l(\mathbf{x}) = F_{l,m-1}(\mathbf{x})\}_1^K} = y_{ik} - p_{k,m-1}(\mathbf{x}_i) \tag{3.3}$$

where $p_{k,m-1}(\mathbf{x})$is derived from $F_{k,m-1}(\mathbf{x})$through the equation 3.3.

After finding out the search direction, one tree is constructed at each iteration m for class k to predict the pseudoresponse $\{\tilde{y}_{ik}\}_{i=1}^N$ by least-squares. Each tree has J-terminal nodes, creating regions $\{R_{jkm}\}_{j=1}^J$ that are defined by the terminal nodes of the tree. The optimal search coefficients in 'line search' can then be found by solving:

$$\{\gamma_{jkm}\} = \arg\min_{\{\gamma_{jk}\}} \sum_{i=1}^{N} \sum_{k=1}^{K} \phi\left(y_{ik}, F_{k,m-1}(\mathbf{x}_i) + \sum_{j=1}^{J} \gamma_{jk} \mathbf{1}(\mathbf{x}_i \in R_{jm})\right) \tag{3.4}$$

where $\mathbf{1}(.)$ is the indicator function that has value 1 if its argument is true, and 0 otherwise. $\phi(y_k, F_k) = y_k \log p_k$ is derived from equation 3.1.

This problem has no closed-form solution. The solution is approximated with a single Newton-Raphson step, using a diagonal approximation to the Hessian, which decomposes the problem into separate calculations for each terminal node of each tree and gives the result:

$$\gamma_{jkm} = \frac{K-1}{K} \frac{\sum_{\mathbf{x}_i \in R_{jkm}} \tilde{y}_{ik}}{\sum_{\mathbf{x}_i \in R_{jkm}} |\tilde{y}_{ik}| \left(1 - |\tilde{y}_{ik}|\right)}, j = 1, 2...J \tag{3.5}$$

3.2.3. Model Evaluation

During the evaluation phase, the recording is traversed using a sliding window of the same length (1 second or 500 samples) as during training but with a different overlap (80%, 200 ms). The ensemble classifier takes the mean values of twelve joint kinematics within each window as the input and returns the recognized activity type for that specific interval by summing the output of each base classifier as Figure 3.3 shows.

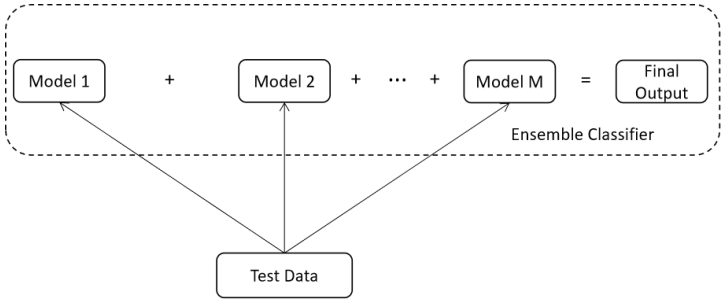


Figure 3.3: Test phase of Gradient Boosting Decision Trees.

Ideally, the window would move forward one timestep at a time. However, this approach would result in significant processing time, which is unsuitable for a near real-time application. To address this issue, a step size of 75 samples (150ms) is used between windows. This compromise strikes a balance between having more detailed predictions with a smaller step size, but longer evaluation time, and a larger step size that reduces processing time at the cost of some prediction detail.

Nevertheless, this introduces a challenge: there are fewer predictions than timesteps in the recording. Consequently, postprocessing is necessary to align the predictions with the recording. One simple way to do this is the interpolation postprocessing. It is defined as all the timesteps between the end of window i and window $i+1$ are assigned to the prediction of window $i+1$.

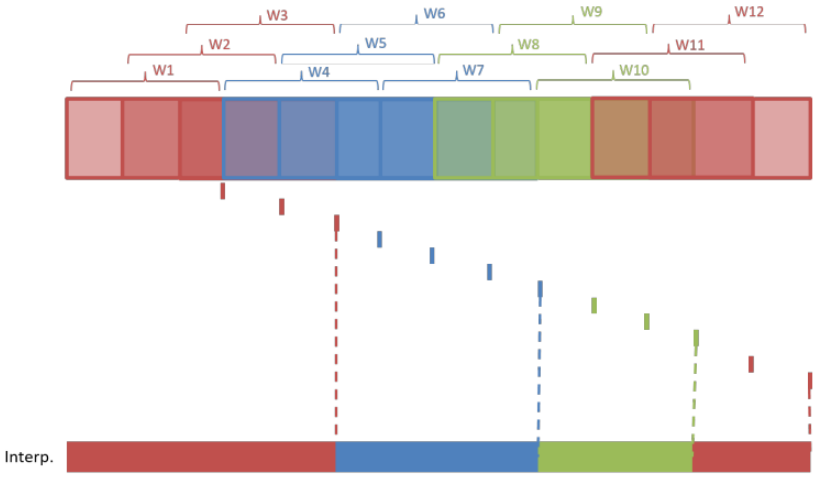


Figure 3.4: Interpolate postprocessing option [29].

Figure 3.4 explains this process for a three-class classification problem. The horizontal multicolored

bar at the top of the figure represents the recording. The windows used for classification are depicted as rectangles with thick borders. In the middle of the figure, the prediction made by each window is represented by a color (red, blue, or green). The bottom horizontal bar displays the resulting predictions after postprocessing. This postprocessing option is very intuitive and fast and easy to compute; however, it does not "clean" undesired and isolated predictions, as it will be shown later.

Since the Gradient Boosting Decision Trees (GBDT) classifier can directly output the probability for each class, the Best score postprocessing technique [29] can be employed to solve the above issue. This technique involves assigning all the timesteps within a window to the prediction made by that window. The final prediction for each timestep is determined by selecting the prediction with the highest probability or confidence among all the windows encompassing that particular.

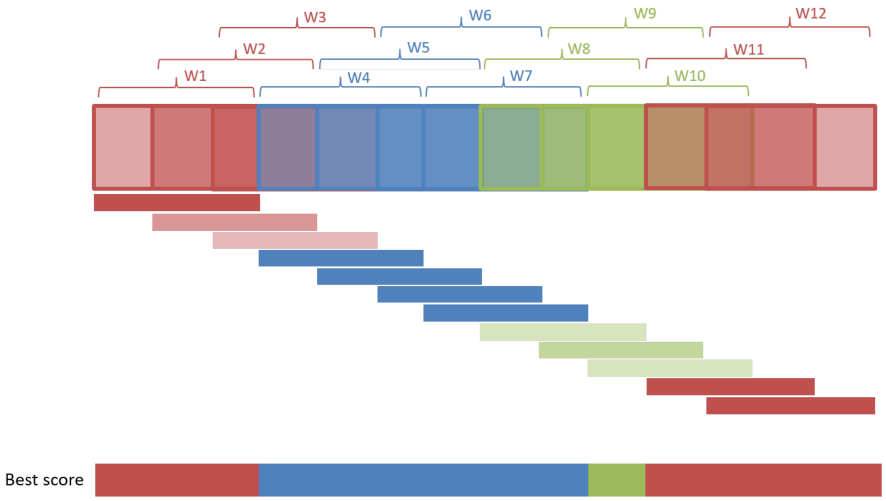


Figure 3.5: Best score postprocessing option [29].

Figure 3.5 illustrates this process for the same three-class classification problem, while the tone of the color represents the confidence level (light to dark). The bottom horizontal bar displays the resulting predictions, where the highest confidence prediction is considered the corresponding timestep's final prediction.

For implementation purposes, a full zero matrix of size $K \times N$ (where K represents the class number and N is the total number of timesteps) is constructed. Additionally, a submatrix with a size of $K \times 500$ is created for the prediction result of each window. Each row of this submatrix contains 500 copies of the predicted probability for the corresponding class. The submatrix is then added to the output matrix at the location corresponding to the timesteps within the current window. Finally, the maximum value in each column of the output matrix is selected to obtain the final prediction for each timestep.

Before generating the final outcome, a procedure for eliminating outliers is implemented to address the possibility of isolated short-duration peaks persisting. If an activity is detected to last less than 150 milliseconds, such a prediction is unrealistic and needs to be removed as an outlier. The final step of the postprocessing method involves taking the mode of the predicted labels within each 150 ms step (75 samples) and assigning the majority label as the final output for that interval. This postprocessing option ensures a fast, scalable, and memory-friendly approach.

3.3. Result[1]

In this section, results obtained with the proposed activity recognition model on the training set and the test set (see Appendix B) will be presented. An early-stopping strategy is applied to avoid overfitting during the training phase. The model training will be terminated if the model's cross-entropy loss doesn't decrease within 10 iterations. Figure 3.6 shows the model loss during the training phase. The maximum iteration is set as 900 and the model stopped training at the 147^{th} iteration.

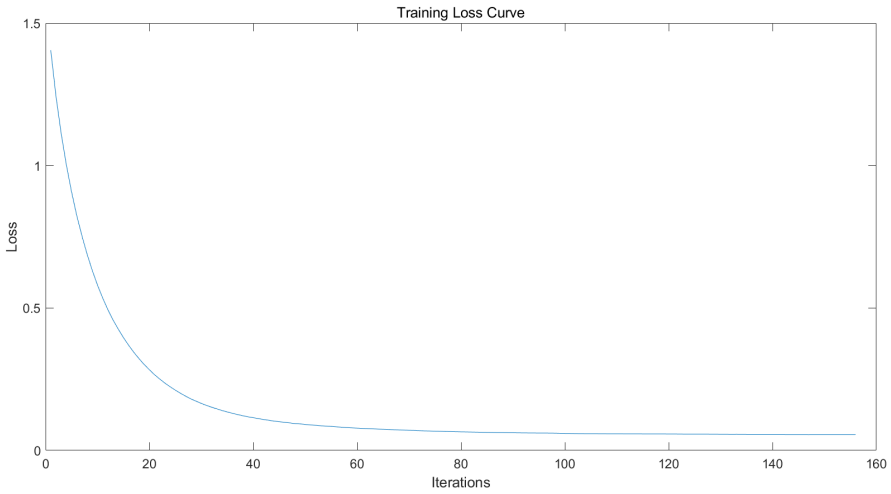


Figure 3.6: Model training loss (Multi-class logistic loss).

To evaluate the performance of the derived model, the confusion matrix of the prediction results on the test set is shown in Figure 3.7.

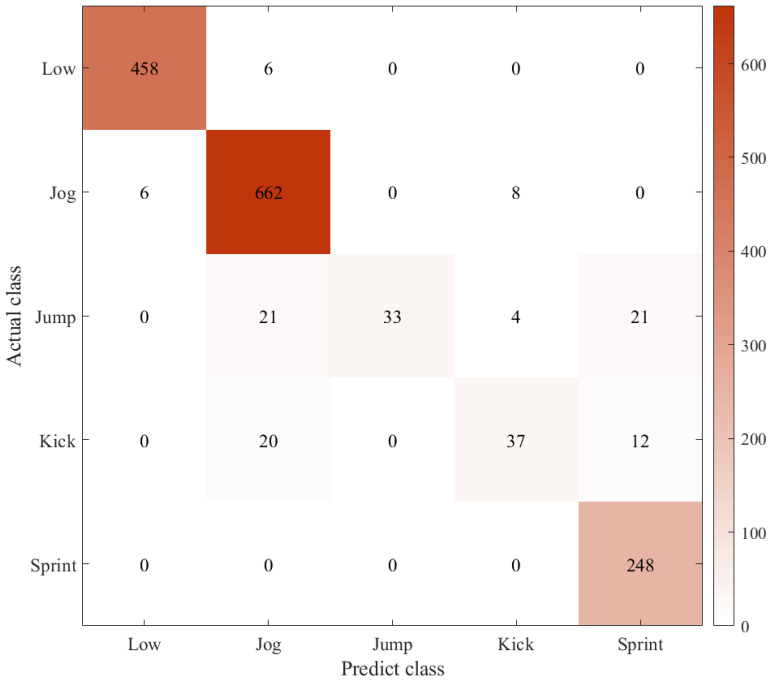


Figure 3.7: Confusion matrix of the proposed model's test results.

[1]The experimental results in this section are derived using the data recorded for Subject 3 in the provided dataset. The detailed specification of the used data can be found in appendix B.

Furthermore, the F1-score per-class (one-vs-rest manner) and overall accuracy of test set results are calculated from the above confusion matrix and given in Table 3.2. The overall accuracy for a multi-class classifier is defined as the ratio of correctly classified data points out of all the samples:

$$\text{Accuracy} = \frac{\text{Number of correctly classified samples}}{\text{Number of total samples}} \tag{3.6}$$

F-1 score is a metric that can symmetrically represent the prediction precision and sensitivity of the classifier for class i. It is defined as the harmonic mean of the precision and recall:

$$\text{F-1 Score}(\text{class} = i) = \frac{2 \times \text{Precision}(\text{class} = i) \times \text{Recall}(\text{class} = i)}{\text{Precision}(\text{class} = i) + \text{Recall}(\text{class} = i)} \tag{3.7}$$

The $\text{Precision}(\text{class} = i)$ and $\text{Recall}(\text{class} = i)$ are calculated as:

$$\begin{aligned} \text{Precision}(\text{class} = i) &= \frac{\text{True positive}(\text{class} = i)}{\text{True positive}(\text{class} = i) + \text{False positive}(\text{class} = i)} \\ \text{Recall}(\text{class} = i) &= \frac{\text{True positive}(\text{class} = i)}{\text{True positive}(\text{class} = i) + \text{False negative}(\text{class} = i)} \end{aligned} \tag{3.8}$$

where True positive (TP) is the number of samples where the model correctly predicts as class i, False positive (FP) is the number of samples where the model incorrectly predicts as class i, and False negative (FN) is the number of samples where the model incorrectly predicts as other classes.

Table 3.2: F-1 scores and accuracy of the prediction results given by the proposed model.

F-1$_{Low}$	F-1$_{Jog}$	F-1$_{Jump}$	F-1$_{Kick}$	F-1$_{Sprint}$	Overall Accuracy
0.9871	0.9600	0.5893	0.6271	0.9376	0.9362

To compare the effect of different postprocessing methods, the prediction results for Subject 3 on the final training rounds (test data set) with two postprocessing methods are shown in Figure 3.8 and Figure 3.9. It can be seen from the graphs that compared to interpolation postprocessing, the best score postprocessing techniques can effectively eliminate undesirable short peaks in the model's output, resulting in more accuracy and smoother prediction results. On the other hand, it can also be observed that the model failed to capture a kicking movement in the first test and a jumping activity in the second test, as the previous discussion suggested.

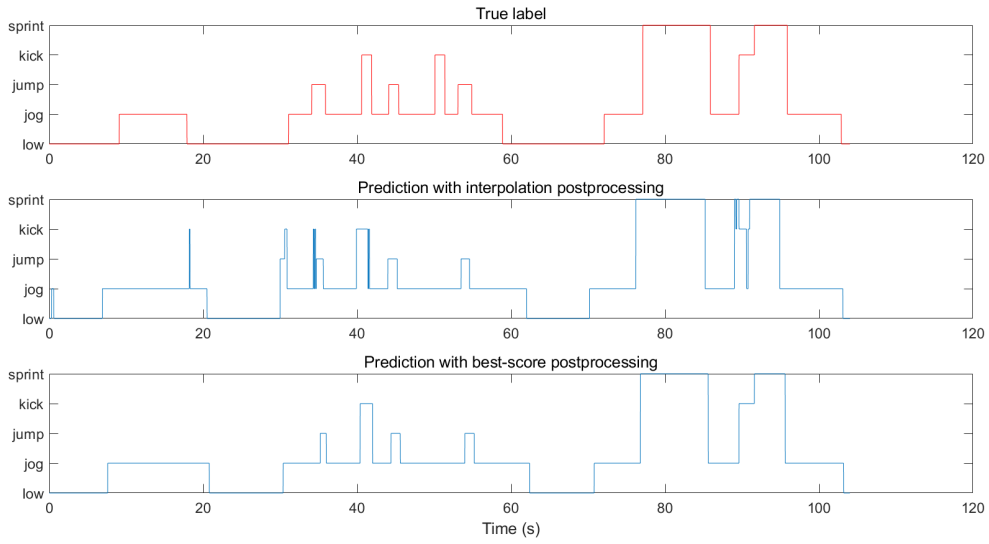


Figure 3.8: Prediction result for Subject 3 during the last training round using different postprocessing methods (first half).

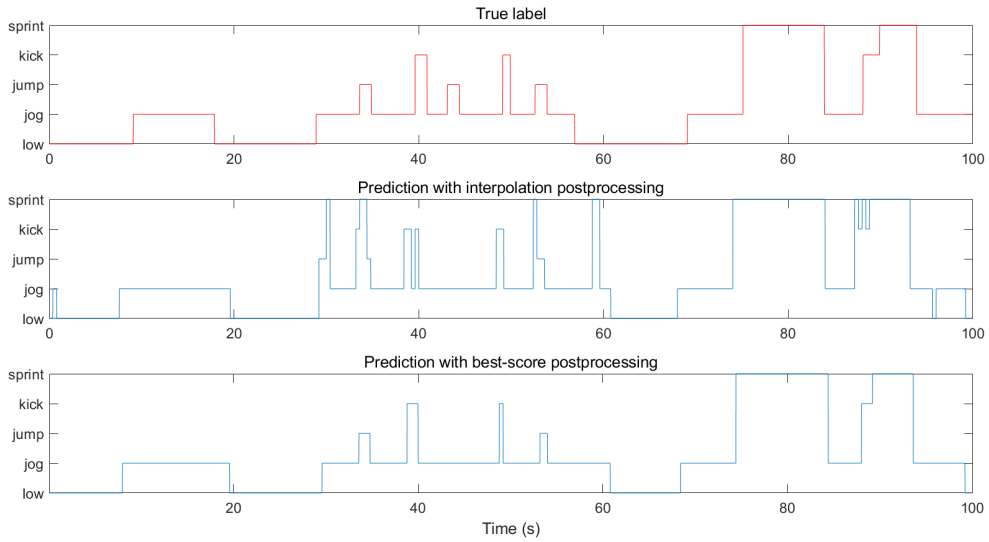


Figure 3.9: Prediction result for Subject 3 during the last training round using different postprocessing methods (second half).

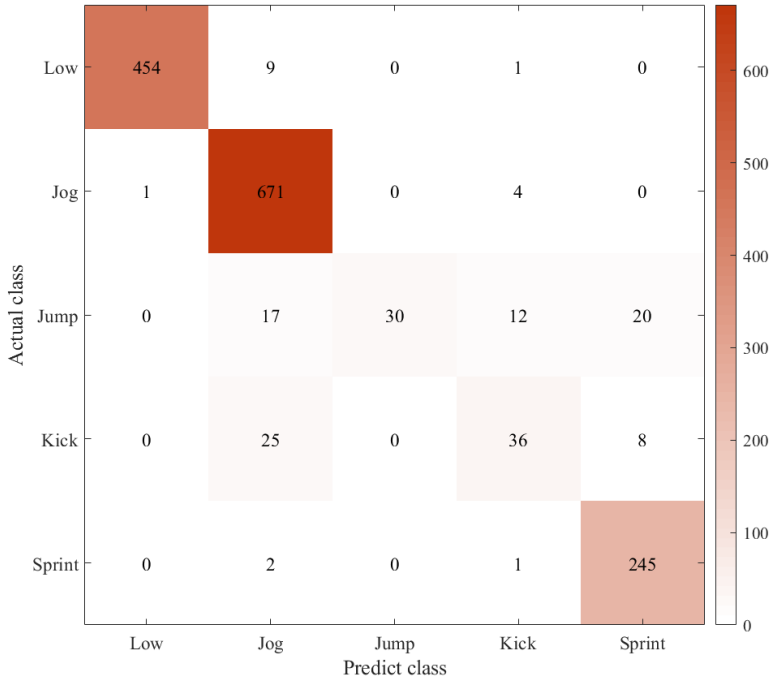


Figure 3.10: Confusion matrix of the test results given by the model trained on reference [10] joint kinematics.

3.4. Discussion

From Table 3.2 and the confusion matrix in Figure 3.7, it can be observed that the classifier performs well on movements that last for a period of time (i.e. walking, jogging, and sprinting) but is less sensitive to explosive movements (i.e. jumping and kicking). The primary reason for this issue is that the model takes the mean value of the joint kinematics within the window as the input, which can make the model sometimes ignore rapid changes in joint kinematics that often occur during explosive activities. One way to solve this issue is to use windows with a smaller size; however, this will introduce unwanted outliers in the prediction. A better way is to find features that can reflect the change of joint kinematics

within the window. Currently, features including interquartile range (IQR), third quartile, and variance are tried, but their corresponding models have little difference in performance from the original model.

For jumping movement, the main source of error arises from where certain samples are misclassified as sprinting. This can be attributed to the player engaging in jumping actions for heading purposes during training, necessitating an acceleration phase before the jump to generate sufficient power for the header. For Kicking movement, the main error occurs when some samples are wrongly classified as jogging. This misclassification stems from the predominantly clandestine nature of passing during the training, during which the player was engaged in jogging.

Furthermore, to investigate the effect of the sensor calibration error on the performance of the football activity recognition model, a model was trained and tested using the joint kinematics data derived on the same data set, with the same training-test set division, but from the reference [10] calibration parameters. The resultant confusion matrix and performance are given in Figure 3.10 and Table 3.3 respectively.

Table 3.3: F-1 scores and accuracy of the prediction results given by the model trained on reference [10] joint kinematics.

F-1$_{Low}$	F-1$_{Jog}$	F-1$_{Jump}$	F-1$_{Kick}$	F-1$_{Sprint}$	Overall Accuracy
0.9880	0.9586	0.5505	0.5854	0.9405	0.9349

By comparing Table 3.2 and Table 3.3, it can be seen that the classifier is more sensitive to the sensor calibration error on explosive movements (with a performance difference of 0.0388 for jumping and 0.0417 for kicking) compared to periodic movements (with a performance difference of 0.0009 for low-intensity activity, 0.0014 for jogging, and 0.0029 for sprinting), which is consistent with the issue discussed in the previous chapter that the joint kinematics estimation for high-intensity activities is most affected by the sensor calibration error. Meanwhile, it can also be noticed that the general performance of the two models is not significantly different (with a 0.0013 difference in overall accuracy), suggesting the proposed model is robust to the sensor calibration error.

4

System Integration and Data Visualization

In this chapter, a near real-time data processing system for the Smart Sensor Shorts is designed and implemented. This system integrates the automatic sensor calibration and activity recognition that were previously introduced. Furthermore, in order to enhance interactivity and improve the user experience, a user interface is developed to display the player's joint kinematics, biomechanical load, activity statistics, and movement in real-time.

4.1. System Integration

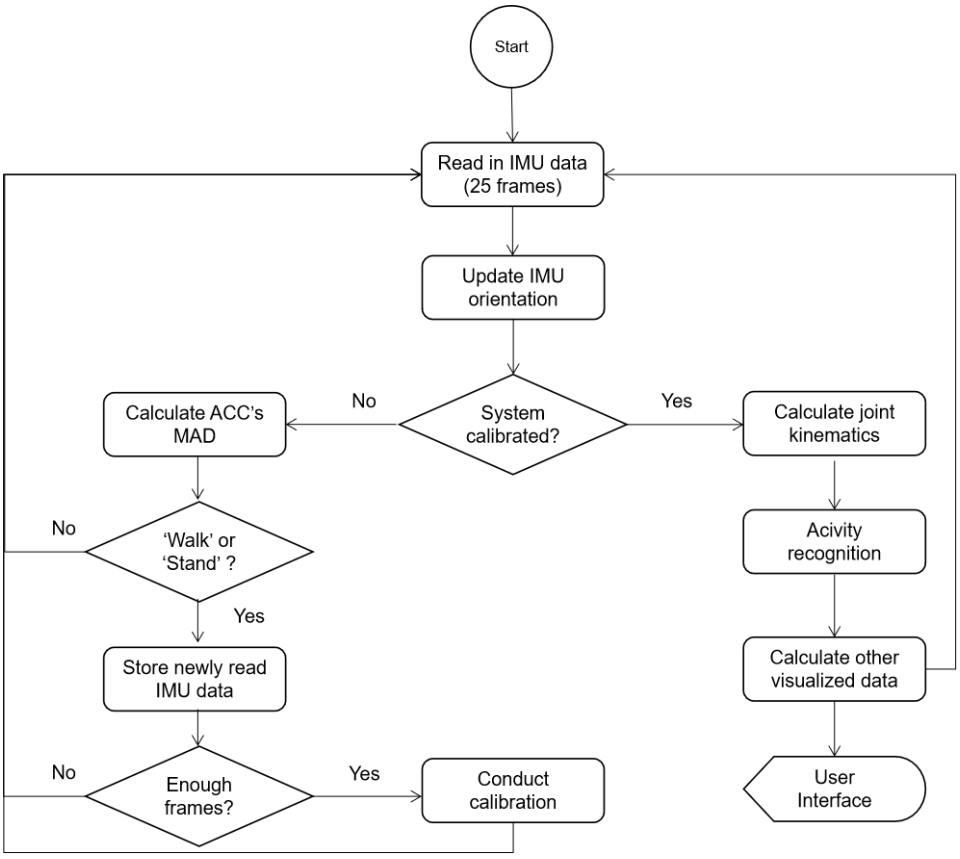


Figure 4.1: System flowchart.

The designed system is shown in Figure 4.1. The system process 25 data frames per iteration. This is because the sampling frequency of the inertial-based tracking system [10] is $500\,Hz$, and the displaying frame rate of the user interface is set to $20\,Hz$ to ensure a smooth transition between each update.

When implementing the system, a moving average approach is employed as Figure 4.2 shows. Since both the mean amplitude deviation (MAD) for sensor calibration and the mean joint kinematics for activity recognition are calculated in windows of data, each with a size of 500 samples, the raw accelerometer, gyroscope, and magnetometer data for each sensor, as well as the angles and angular velocities for each joint, are all stored in 3×500 matrices within the system to save the memory space. Whenever new data is received, the system will discard the oldest 25 data points and append the newly derived 25 data points at the end of the existing data matrix.

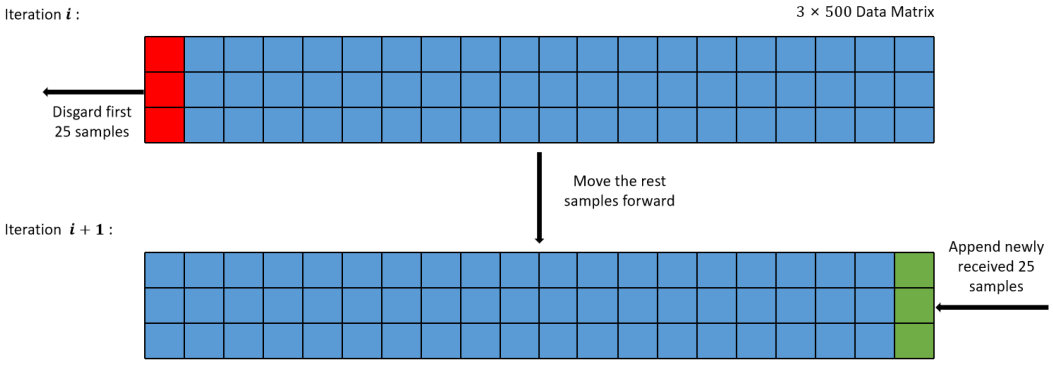


Figure 4.2: Diagram of data storage in the system.

During one iteration, the system will begin by updating the orientation of three sensors based on the newly obtained raw IMU data. Then, the system will check if all sensors are properly calibrated relative to their corresponding body segment frames. If the calibration is incomplete, the system will identify the periods of standing and walking, and subsequently store the IMU readouts during these periods for future calibration purposes. Otherwise, the system will proceed to compute the angles and angular velocities of the hip joints, determine the current activity type, and calculate the physical load of the hip joints (which will be introduced in the subsequent section). Note that, due to a step size of 100 samples ($200\,ms$) between each window in the activity recognition model, the activity recognition process is not executed for every iteration but rather every 4 iterations.

4.2. Physical Load Quantification

Hamstring strain injuries result from a disruption in the muscle fibers of one of the hamstring muscles. As highlighted in Chapter 1, there has been a substantial increase in the occurrence and impact of hamstring injuries in men's professional football over the past two decades, nearly doubling in frequency [30]. Notably, these injuries account for approximately 24% of all injuries observed in professional football, and a previous hamstring injury can pose a great risk for subsequent hamstring injuries [31]. Consequently, it is imperative to reflect the physical load of the hamstring using the Smart Sensor Short to help the player and medical staff prevent such injury from the outset.

4.2.1. Biomechanical Load

To this end, the biomechanical load around the hamstring should be quantified, as the biomechanical load can directly reflect the mechanical stresses and strains on corresponding musculoskeletal tissues. Bastiaansen et al. [32] introduced an indicator in their work recently to estimate the biomechanical load for the lower extremities using the segment angular accelerations derived from an inertial sensor setup. The biomechanical load around each joint is quantified as:

$$\text{Load} = \frac{\left(\sqrt{\left((\alpha_x)^2 + (\alpha_y)^2 + (\alpha_z)^2\right)}\right)^2}{10^8} \tag{4.1}$$

where α_x, α_x, and α_x are joint angular accelerations in x, y, and z axis of the global coordinate system respectively. Within the context of this study, the load of the right and left hips will be calculated, and the hip angular accelerations are derived by differentiating the obtained joint angular velocities.

4.2.2. Neuromuscular Fatigue

Furthermore, repeated findings have consistently demonstrated that a majority of hamstring injuries occur towards the end of each half during competitive matches [3], [30], [33], indicating a potential association between neuromuscular fatigue and the risk of a hamstring injury. Therefore, it is essential to quantify the level of neuromuscular fatigue in the designed system.

Wilmes et al [34] in a recent study assessed the impact of football-specific neuromuscular fatigue on sprinting kinematics in relation to hamstring strength. Their findings indicated a notable decline in hamstring maximal voluntary torque (MVT) throughout a 90-minute simulation of a football match. Furthermore, they discovered a significant correlation between the decrease in hamstring maximal voluntary torque and a decreased peak hip extension (θ_{Hip}).

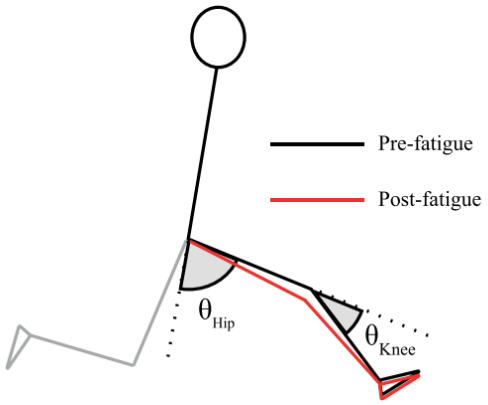


Figure 4.3: Joint angle definitions and effects of fatigue on peak hip flexion. The hip flexion/extension angle (θ_{Hip}) is defined as the angle between the longitudinal axis of the pelvis and the longitudinal axis of the thigh [34].

These outcomes suggest that the hamstrings exhibit reduced capacity in decelerating the lower leg during the final swing phase of the sprint running step cycle, resulting in greater peak hamstring lengths and smaller peak hip angles (Figure 4.3). Therefore, the hip flexion/extension angle (θ_{Hip}) is a reasonable index to reflect the neuromuscular fatigue around the hamstring and is calculated as:

$$\theta_{\text{Hip}} = 180^\circ - \theta_z \tag{4.2}$$

Where θ_z is the rotation angle along the z-axis in degree derived from hip joint orientation quaternion.

4.3. Result

4.3.1. Processing Time

In order to fulfill the near real-time requirement, the system's processing speed must not fall below 20 iterations per second. The system is tested on the last training for Subject 3 [10] and Figure 4.4 illustrates the distribution of processing time per iteration. It can be seen from the figure that the processing time for each iteration remains below 50 milliseconds, indicating that the designed system successfully satisfies the near-real-time criterion.

To further investigate the system performance, the average time per iteration taken by each system part and the whole system is calculated and shown in Table 4.1. It can be seen that updating IMUs' orientation filters and recognizing human activities account for the majority of the time consumed, contributing to 18.30% and 80.54% of the total average frame time respectively.

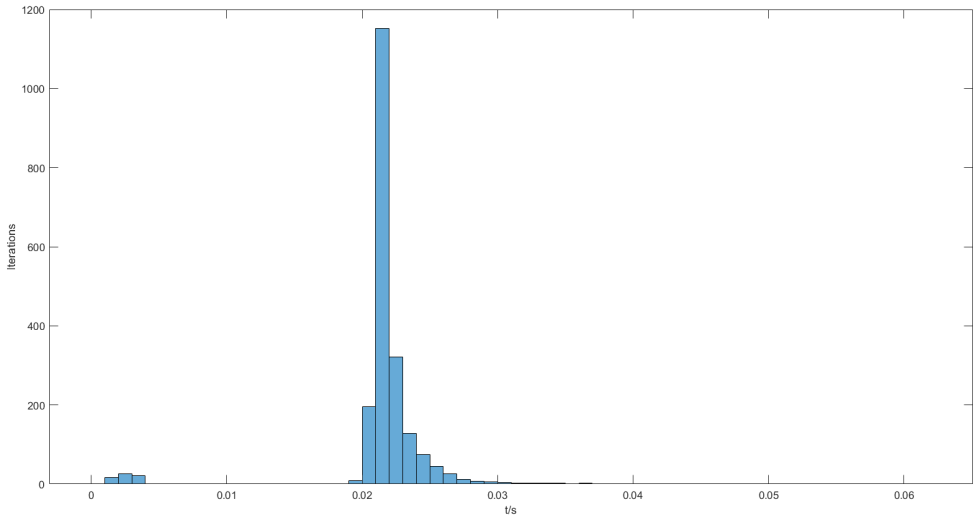


Figure 4.4: Histogram of system processing time per iteration.

Table 4.1: Average processing time (ms) for each system part per iteration.

	IMU Orientation Updating	MAD Calculation	Joint Kinematics Calculation
t/ms	3.93	0.05	0.67
	Activity Recognition	Load Calculation	Overall
t/ms	17.30	0.03	21.48

4.3.2. User Interface

After each system iteration concludes, the processed data is sent to a user interface for visualization. The user interface designed for the Smart Sensor Shorts is shown in Figure 4.5, consisting of 4 sub-windows in total.

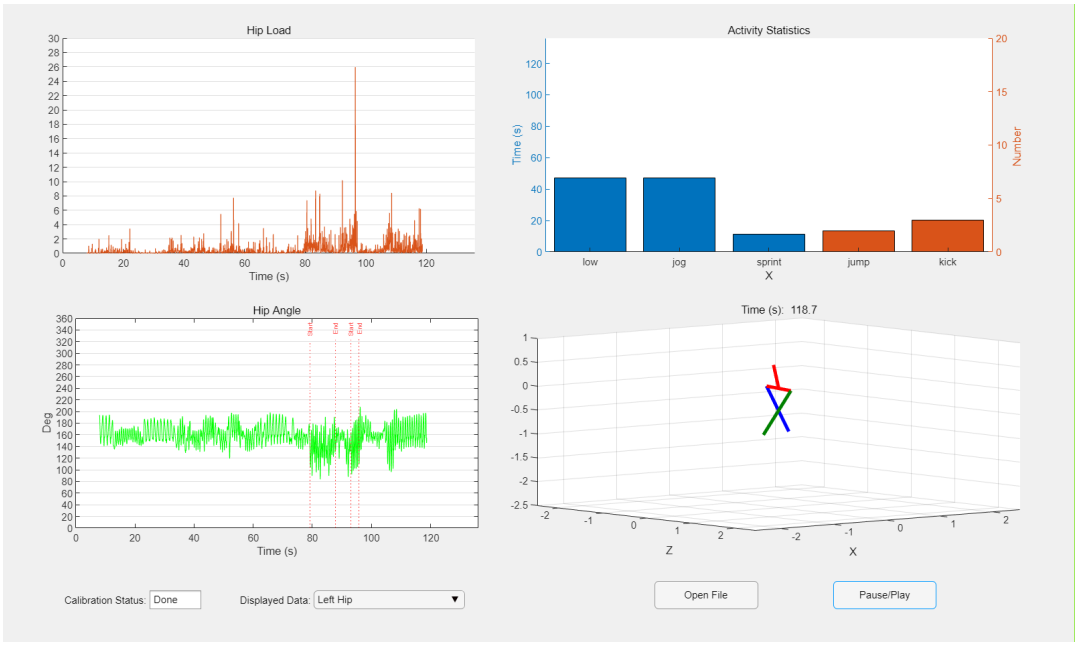


Figure 4.5: User interface for the Smart Sensor Shorts.

The top right graph is a bar plot displaying the time statistics for different types of activities. For activities that are performed repeatedly and periodically (i.e. stand, walk, jog, and sprint), the total duration of each activity so far is counted; and for explosive activities that are performed only once without a repetitive pattern (i.e. kick and jump), the total number of times of each activity are counted.

The bottom right graph is a 3D matchstick model of the human lower limb. It demonstrates the movement of the subject's left and right thighs and pelvis relative to the global coordinate system based on the orientations of three IMUs attached to the corresponding body segment.

Two left graphs show the physical load of the hip joints. The top left graph displays the biomechanical load of hip joints over time. The bottom left graph shows the hip flexion/extension angle over time, and it also specifies the sprint interval derived by the activity recognition module so that the user can better observe the changes in peak hip angles during the sprinting stage. This information can facilitate a better balance between training and recovery, potentially optimizing performance and mitigating the occurrence of injuries.

All the graphs in the user interface are updated 20 times per second. During each update, for the two graphs on the left, 25 newly computed data points are added to the current graph; for the bar plot, $50\,ms$ is added to the current time duration of the corresponding activity if a periodic activity is detected, and 1 is added to the current activity's counts if an explosive activity is detected; for the 3D model, the orientation of each segment is updated according to the last estimated IMU quaternion.

At the bottom of the interface, the user can choose to display the data of the right/left hip in two left graphs using the drop-down button. The calibration status of the system is shown in the text area and the system can be paused using the Pause/Play button.

5

General Discussion

This thesis comprised a series of experiments to build a near-real-time data analysis system for the Smart Sensor Short. Chapter 2 explored an automatic calibration method for IMU sensors attached to human lower extremities. Chapter 3 developed a gradient-boosted decision tree model to recognize football-specific activity based on lower body joint angles and joint angular velocities. Chapter 4 built a system combining functions developed in previous chapters and designed a user interface to demonstrate the motion and loads of the user's lower limb. The implications and potential future work of these studies will be discussed in this chapter.

5.1. Sensor-to-Body Calibration of IMUs

In this thesis, an automatic IMU sensor calibration technique based on standing/walking detection is developed, achieving an $18.92°(\pm 5.74°)$ average calibration error for each sensor. Although the calibration accuracy obtained by the proposed method is comparable with most of the suggested calibration-specific movements/ postures, to enable a more widespread adoption of IMU-based motion analysis systems, several things in the current method can be further improved.

First of all, the second calibration stage of the pelvis sensor requires the calibration results of the other two sensors to estimate the direction of the horizontal component of the geomagnetic field and relies on the three sensors' magnetometer readout. Consequently, the calibration accuracy of the pelvis sensor is easily affected by the calibration accuracy of the other two sensors and becomes vulnerable to the potential interference caused by external magnetic fields. This problem can be solved by integrating and wiring the pelvis sensor into clothing as shown by Steijlen et al. [9] to fix its orientation to the body trunk. Furthermore, the proposed sensor-to-body calibration method was investigated for only three able-bodied participants and should be further investigated with more participants, specifically for individuals with various movement disorders, which might violate the main hypothesis of the proposed calibration, i.e., single-axis rotation of lower limb in the sagittal plane during straight walking. Last but not least, hitting the ground while walking in real training will introduce additional noise to the gyroscope readout due to soft tissue artifacts compared to the experiment setup, affecting the accuracy of the principal component analysis; therefore a filter to reduce soft tissue artifacts can be further developed.

5.2. Football-specific Activity Recognition

In this thesis, a gradient-boosted tree model was built and trained to identify different types of activities in football training based on two hips' joint angles and joint angular velocities. Although it achieves a 93.62% overall accuracy, its performance on kicking and jumping movements is not ideal due to insufficient training samples. There exists a severe class imbalance in the current dataset, as the player spent less than 10% of the total training time on kicking and jumping. Therefore it is recommended to acquire more joint kinematics data for these two types of activities to improve the classifier's performance. Nevertheless, it should be noticed here that the misclassification of jumping and kicking will not affect the ultimate goal of the system - to monitor the user load. This is because the joint load is

calculated from the joint angle and angular velocity, whose accuracy depends on the result of sensor calibration and is independent of the HAR model output. At the same time, since the model can accurately predict the sprinting movement, the user can still use the system to observe the change in the joint angle during the sprint period.

Also, currently, only the mean value of the joint angles and joint angular velocities within the sliding window is taken as the model input. The built-in smoothing effect of averaging may cancel out rapid or small changes in the joint data, making it more difficult to detect explosive movements. Thus, other higher-level features of joint kinematics can be further extracted to reflect the extent or speed of change within the data. Lastly, considering the current developments in the areas of deep learning, a deep-learning-based method can be employed here to obtain a better result. Although the current dataset has insufficient data to build a deep network, several large open source datasets [35], [36] consisting of daily human activities could be used to build a base model, and then use transfer learning techniques to fine-tune it with football data to allow it to recognize specific football activities.

5.3. System Design

This thesis presents the design of a near-real-time data analysis system for the Smart Sensor Short. The system integrates automatic sensor calibration, football activity recognition, and training load assessment functions, and achieves an average processing speed of 21 ms per iteration. The system also includes a user interface that displays lower body movement, lower limb load, and activity statistics at a rate of 20 frames per second (FPS). However, there are several areas where the current system can be further improved.

To fulfill the near-real-time requirement, the data processing and data displaying are currently executed in two separate files, as plotting in Matlab is time-consuming. In the future, it would be beneficial to develop a more efficient method for visualizing the processed data if simultaneous data processing and monitoring are desired by the user. Additionally, the current system only considers single-joint kinematics when quantifying training load. To enhance the insights provided, it is advisable to incorporate information from multiple joints instead of relying on a single joint. Furthermore, it is recommended to investigate the relationship between the biomechanical load metrics derived by the system and injuries. This analysis would enable the system to early detect and even prevent injuries.

While the IMU-based motion tracking system used in this study does not report any significant error due to sensor drift, and the sensor fusion algorithm [17] can effectively correct the integration drift of gyroscope data using accelerometer and magnetometer data when estimating orientations, it is still necessary to include a sensor drift checking unit in the system. This is particularly important because the training load estimation relies solely on accelerometer data, which can be susceptible to sensor drift errors. Therefore, implementing a checking unit for sensor drift would enhance the accuracy and reliability of the system.

6

Conclusion

This thesis proposes a near real-time data analysis system for Smart Sensor Shorts, designed to monitor the movement and load of users' lower limbs during football training. The system comprises an automatic sensor calibration module, a lower joint kinematics calculation module, a football-specific action recognition module, a load calculation module, and a user interface.

Chapter 2 introduces an automatic sensor-to-body calibration algorithm based on standing and walking detection. This algorithm calculates the mean amplitude deviation of raw accelerometer readouts fixed at the user's pelvis to identify instances of still standing and straight walking during training. By aligning the sensor reference frame with the corresponding body segment frame using the single-axis feature of the movement, the algorithm enables subsequent calculations of joint kinematics. Unlike traditional calibration algorithms, the proposed method eliminates the need for users to perform calibration-specific actions and does not require precise timing. This significantly simplifies the calibration process while maintaining a high calibration accuracy ($18.92° \pm 5.74°$ average calibration error at each sensor location), thereby ensuring the widespread usability of the system.

Chapter 3 presents a football-specific activity recognition model based on gradient-boosting decision trees. Once the system's joint kinematics calculation module determines the hip joint angle and angular velocity using the calibration parameters and IMU orientation quaternion, the model predicts the user's current activity (low-intensity movement, jogging, jumping, kicking, or sprinting). These predictions help users track the timing and frequency of different activities during training, particularly in identifying sprinting periods as changes in the peak value of each lower limb joint angle during sprinting reflect fatigue levels. The proposed model not only guarantees comparable accuracy (93.62% overall accuracy) to current mainstream deep learning human activity recognition methods but also reduces the amount of training data required due to its simple structure. Additionally, it provides faster evaluation times, making it more suitable for real-time systems.

Chapter 4 details the pipeline of the proposed near-real-time data processing system. The system stores the collected sensor data in fixed-sized containers (500×3) using a moving average ideology to conserve memory space. In each system iteration, the load calculation module derives joint angular acceleration by differentiating the previously obtained angular velocity and calculates the corresponding joint load. Furthermore, Chapter 4 introduces the designed user interface for the system. The user interface features four graphs: a bar graph displaying the activity recognition results, a 3D match-stick graph visualizing lower body movement, a line graph illustrating the load on hip joints over time, and another line graph depicting the peak hip flexion angle changes during different sprint intervals. Experimental results demonstrate that the system meets the near real-time requirement with a processing speed of 21 ms per iteration, and the user interface maintains a stable data display at 20 FPS.

References

[1] J. J. Malone, R. Di Michele, R. Morgans, D. Burgess, J. P. Morton, and B. Drust, "Seasonal training-load quantification in elite english premier league soccer players," *International journal of sports physiology and performance*, vol. 10, no. 4, pp. 489–497, 2015.

[2] A. D. White and N. G. MacFarlane, "Analysis of international competition and training in men's field hockey by global positioning system and inertial sensor technology," *The Journal of Strength & Conditioning Research*, vol. 29, no. 1, pp. 137–143, 2015.

[3] J. Ekstrand, M. Hägglund, and M. Waldén, "Epidemiology of muscle injuries in professional football (soccer)," *The American journal of sports medicine*, vol. 39, no. 6, pp. 1226–1232, 2011.

[4] J. H. Stubbe, A.-M. M. Van Beijsterveldt, S. Van Der Knaap, *et al.*, "Injuries in professional male soccer players in the netherlands: A prospective cohort study," *Journal of athletic training*, vol. 50, no. 2, pp. 211–216, 2015.

[5] P. Wong and Y. Hong, "Soccer injury in the lower extremities," *British journal of sports medicine*, vol. 39, no. 8, pp. 473–482, 2005.

[6] J. Ekstrand, M. Waldén, and M. Hägglund, "Hamstring injuries have increased by 4% annually in men's professional football, since 2001: A 13-year longitudinal analysis of the uefa elite club injury study," *British journal of sports medicine*, vol. 50, no. 12, pp. 731–737, 2016.

[7] C. Barnes, D. Archer, B. Hogg, M. Bush, and P. Bradley, "The evolution of physical and technical performance parameters in the english premier league," *International journal of sports medicine*, vol. 35, no. 13, pp. 1095–1100, 2014.

[8] K. Inoue, H. Nunome, T. Sterzing, H. Shinkai, and Y. Ikegami, "Dynamics of the support leg in soccer instep kicking," *Journal of Sports Sciences*, vol. 32, no. 11, pp. 1023–1032, 2014.

[9] A. Steijlen, B. Burgers, E. Wilmes, *et al.*, "Smart sensor tights: Movement tracking of the lower limbs in football," *Wearable Technologies*, vol. 2, 2021.

[10] E. Wilmes, C. J. de Ruiter, B. J. Bastiaansen, *et al.*, "Inertial sensor-based motion tracking in football with movement intensity quantification," *Sensors*, vol. 20, no. 9, p. 2527, 2020.

[11] D. M. Kelly, W. Gregson, T. Reilly, and B. Drust, "The development of a soccer-specific training drill for elite-level players," *The Journal of Strength & Conditioning Research*, vol. 27, no. 4, pp. 938–943, 2013.

[12] M. Nazarahari and H. Rouhani, "Semi-automatic sensor-to-body calibration of inertial sensors on lower limb using gait recording," *IEEE Sensors Journal*, vol. 19, no. 24, pp. 12 465–12 474, 2019.

[13] H. Vähä-Ypyä, T. Vasankari, P. Husu, J. Suni, and H. Sievänen, "A universal, accurate intensity-based classification of different physical activities using raw data of accelerometer," *Clinical physiology and functional imaging*, vol. 35, no. 1, pp. 64–70, 2015.

[14] H. Vähä-Ypyä, T. Vasankari, P. Husu, *et al.*, "Validation of cut-points for evaluating the intensity of physical activity with accelerometry-based mean amplitude deviation (mad)," *PloS one*, vol. 10, no. 8, e0134813, 2015.

[15] M. Aittasalo, H. Vähä-Ypyä, T. Vasankari, P. Husu, A.-M. Jussila, and H. Sievänen, "Mean amplitude deviation calculated from raw acceleration data: A novel method for classifying the intensity of adolescents' physical activity irrespective of accelerometer brand," *BMC sports science, medicine and rehabilitation*, vol. 7, pp. 1–7, 2015.

[16] G. Wu, P. R. Cavanagh, *et al.*, "Isb recommendations for standardization in the reporting of kinematic data," *Journal of biomechanics*, vol. 28, no. 10, pp. 1257–1262, 1995.

[17] S. O. Madgwick, A. J. Harrison, and R. Vaidyanathan, "Estimation of imu and marg orientation using a gradient descent algorithm," in *2011 IEEE international conference on rehabilitation robotics*, IEEE, 2011, pp. 1–7.

[18] G. Wu, S. Siegler, P. Allard, *et al.*, “Isb recommendation on definitions of joint coordinate system of various joints for the reporting of human joint motion—part i: Ankle, hip, and spine,” *Journal of biomechanics*, vol. 35, no. 4, pp. 543–548, 2002.

[19] M. Nazarahari, A. Noamani, N. Ahmadian, and H. Rouhani, “Sensor-to-body calibration procedure for clinical motion analysis of lower limb using magnetic and inertial measurement units,” *Journal of Biomechanics*, vol. 85, pp. 224–229, 2019.

[20] G. Kaketsis, “Classification in football: Activity classification using sensor data in football,” 2020.

[21] D. Schuldhaus, C. Zwick, H. Körger, E. Dorschky, R. Kirk, and B. M. Eskofier, “Inertial sensor-based approach for shot/pass classification during a soccer match,” in *KDD workshop on large-scale sports analytics*, 2015, pp. 1–4.

[22] T. Kautz, B. H. Groh, J. Hannink, U. Jensen, H. Strubberg, and B. M. Eskofier, “Activity recognition in beach volleyball using a deep convolutional neural network: Leveraging the potential of deep learning in sports,” *Data Mining and Knowledge Discovery*, vol. 31, pp. 1678–1705, 2017.

[23] L. Wang and R. Liu, “Human activity recognition based on wearable sensor using hierarchical deep lstm networks,” *Circuits, Systems, and Signal Processing*, vol. 39, pp. 837–856, 2020.

[24] C. Xu, D. Chai, J. He, X. Zhang, and S. Duan, “Innohar: A deep neural network for complex human activity recognition,” *Ieee Access*, vol. 7, pp. 9893–9902, 2019.

[25] K. Xia, J. Huang, and H. Wang, “Lstm-cnn architecture for human activity recognition,” *IEEE Access*, vol. 8, pp. 56 855–56 866, 2020.

[26] L. Breiman, J. Friedman, R. Olshen, and C. Stone, “Cart,” *Classification and regression trees*, 1984.

[27] J. H. Friedman, “Greedy function approximation: A gradient boosting machine,” *Annals of statistics*, pp. 1189–1232, 2001.

[28] M. Jamei, M. Ali, A. Malik, R. Prasad, S. Abdulla, and Z. M. Yaseen, “Forecasting daily flood water level using hybrid advanced machine learning based time-varying filtered empirical mode decomposition approach,” *Water Resources Management*, vol. 36, no. 12, pp. 4637–4676, 2022.

[29] R. Cuperman, K. M. Jansen, and M. G. Ciszewski, “An end-to-end deep learning pipeline for football activity recognition based on wearable acceleration sensors,” *Sensors*, vol. 22, no. 4, p. 1347, 2022.

[30] J. Ekstrand, H. Bengtsson, M. Waldén, M. Davison, K. M. Khan, and M. Hägglund, “Hamstring injury rates have increased during recent seasons and now constitute 24% of all injuries in men’s professional football: The uefa elite club injury study from 2001/02 to 2021/22,” *British Journal of Sports Medicine*, vol. 57, no. 5, pp. 292–298, 2023.

[31] B. Green, M. N. Bourne, N. van Dyk, and T. Pizzari, “Recalibrating the risk of hamstring strain injury (hsi): A 2020 systematic review and meta-analysis of risk factors for index and recurrent hamstring strain injury in sport,” *British Journal of Sports Medicine*, vol. 54, no. 18, pp. 1081–1088, 2020.

[32] B. J. Bastiaansen, R. J. Vegter, E. Wilmes, C. J. de Ruiter, K. A. Lemmink, and M. S. Brink, “Biomechanical load quantification using a lower extremity inertial sensor setup during football specific activities,” *Sports biomechanics*, pp. 1–16, 2022.

[33] C. Woods, R. Hawkins, S. Maltby, M. Hulse, A. Thomas, and A. Hodson, “The football association medical research programme: An audit of injuries in professional football—analysis of hamstring injuries,” *British journal of sports medicine*, vol. 38, no. 1, pp. 36–41, 2004.

[34] E. Wilmes, C. J. De Ruiter, B. J. Bastiaansen, *et al.*, “Associations between hamstring fatigue and sprint kinematics during a simulated football (soccer) match,” *Medicine and science in sports and exercise*, vol. 53, no. 12, p. 2586, 2021.

[35] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, “Activity recognition using cell phone accelerometers,” *ACM SigKDD Explorations Newsletter*, vol. 12, no. 2, pp. 74–82, 2011.

[36] D. Anguita, A. Ghio, L. Oneto, X. Parra, J. L. Reyes-Ortiz, *et al.*, “A public domain dataset for human activity recognition using smartphones.,” in *Esann*, vol. 3, 2013, p. 3.

A

MPU-9150 Product Specification

This appendix provides information regarding the features and electrical specifications of the MPU-9150 Motion Processing Unit working under the typical operating circuit.

A.1. Typical Operating Circuit

The typical operating circuit of MPU-9150 is shown in Figure A.1, where $VLOGIC = 1.8V \pm 5\%$, $VDD = 2.375V$ - $3.465V$, and $T = 25°C$.

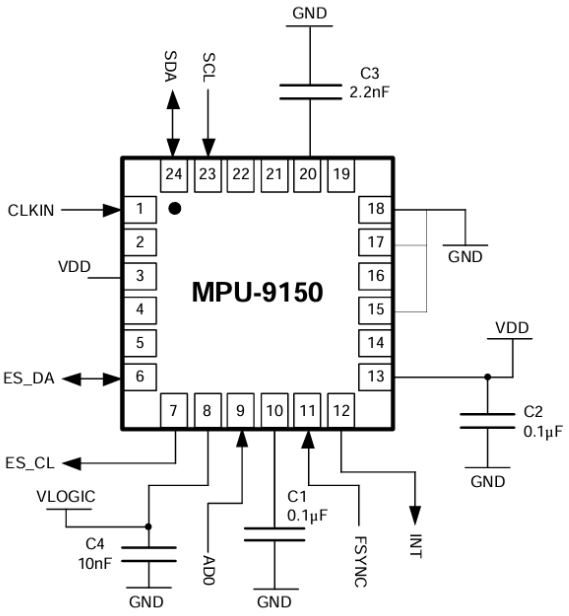


Figure A.1: MPU-9150 typical operating circuit.

A.2. Gyroscope

The triple-axis MEMS gyroscope in the MPU-9150 includes the following features:

- Digital-output X-, Y-, and Z-axis angular rate sensors (gyroscopes) with a user-programmable full-scale range of ±250, ±500, ±1000, and ±$2000\,°/sec$.
- External signal connected to the FSYNC pin supports image, video, and GPS synchronization.
- Integrated 16-bit ADCs enable simultaneous sampling of gyros.
- Improved low-frequency noise performance.

- Digitally-programmable low-pass filter.
- Factory calibrated sensitivity scale factor.

The electrical characteristics of the gyroscope under the typical operating circuit are listed below. The FS_SEL is set to 3 during the test to let the gyroscope measure ±$2000\,^\circ/s$.

Table A.1: Gyroscope Specifications

GYROSCOPE SENSITIVITY	CONDITIONS	TYP	UNITS
Full-Scale Range	$FS\_SEL = 0$	±250	$^\circ/s$
	$FS\_SEL = 1$	±500	$^\circ/s$
	$FS\_SEL = 2$	±1000	$^\circ/s$
	$^*FS\_SEL = 3$	±2000	$^\circ/s$
ADC Word Length		16	bits
Sensitivity Scale Factor	$FS\_SEL = 0$	131	$LSB/(^\circ/s)$
	$FS\_SEL = 1$	65.5	$LSB/(^\circ/s)$
	$FS\_SEL = 2$	32.8	$LSB/(^\circ/s)$
	$^*FS\_SEL = 3$	16.4	$LSB/(^\circ/s)$

A.3. Accelerometer

The triple-axis MEMS accelerometer in MPU-9150 includes the following features:

- Digital-output 3-axis accelerometer with a programmable full-scale range of ±$2\,g$, ±$4\,g$, ±$8\,g$, and ±$16\,g$.
- Integrated 16-bit ADCs enable simultaneous sampling of accelerometers while requiring no external multiplexer.
- Orientation detection and signaling.
- Tap detection.
- High-G interrupt.

The electrical characteristics of the accelerometer under the typical operating circuit are listed below. The AFS_SEL is set to 3 during the test to let the accelerometer measure ±$16\,g$.

Table A.2: Accelerometer Specifications

Accelerometer SENSITIVITY	CONDITIONS	TYP	UNITS
Full-Scale Range	$AFS\_SEL = 0$	±2	g
	$AFS\_SEL = 1$	±4	g
	$AFS\_SEL = 2$	±8	g
	$^*AFS\_SEL = 3$	±16	g
ADC Word Length	Output in two's complement format	16	bits
Sensitivity Scale Factor	$AFS\_SEL = 0$	16384	LSB/g
	$AFS\_SEL = 1$	8192	LSB/g
	$AFS\_SEL = 2$	4096	LSB/g
	$^*AFS\_SEL = 3$	2048	LSB/g

A.4. Magnetometer

The triple-axis MEMS magnetometer in MPU-9150 includes the following features:

- 3-axis silicon monolithic Hall-effect magnetic sensor with a magnetic concentrator.
- Wide dynamic measurement range and high resolution with lower current consumption.
- Output data resolution is 13 bits ($0.3\,\mu T$ per LSB).
- Full-scale measurement range is ±$1200\,\mu T$.
- Self-test function with an internal magnetic source to confirm magnetic sensor operation on end products.

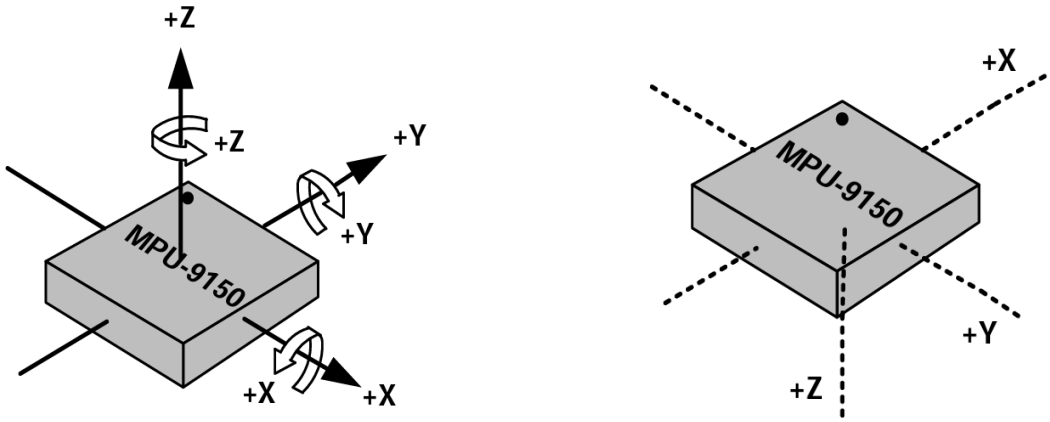
The electrical characteristics of the Magnetometer under the typical operating circuit are listed below.

Table A.3: Magnetometer Specifications

MAGNETOMETER SENSITIVITY	CONDITIONS	TYP	UNITS
Full-Scale Range		±1200	μT
ADC Word Length	Output in two's complement format	13	bits
Sensitivity Scale Factor		0.3	$\mu T/LSB$

A.5. Sensor Frame Definition

The sensor reference frame of the MPU-9150 is defined as follows:



(a) Orientation of axes and polarity of rotation for accelerometer and gyroscope.

(b) Orientation of axes for magnetometer.

Figure A.2: Sensor frame definition.

B

Data Set Specification

The dataset utilized in this thesis was provided by Wilmes et al [10]. The data recorded for Subject 3 serves as the primary source for conducting experiments, as it is the only subject for which video recordings are available. During the collection of the dataset, Subject 3 engaged in four rounds of football training drills. For sensor calibration, the proposed method's performance was assessed across multiple training rounds, and the results were reported by calculating the mean value and standard deviation. In terms of football-specific activity recognition, the model was trained using data from the initial three training rounds and subsequently tested using data from the final training round. To facilitate future replication of these calculations, the starting and ending times of each training round for Subject 3 are specified in the accompanying table.

Table B.1: Training time specification for Subject 3

	Starting Time	Ending Time
First Round	2021-02-18 15:06:03	2021-02-18 15:10:06
Second Round	2021-02-18 15:14:26	2021-02-18 15:18:32
Third Round	2021-02-18 15:22:43	2021-02-18 15:26:42
Fourth Round	2021-02-18 15:30:36	2021-02-18 15:34:37

C

Sensor Calibration Result for Subject 1 and Subject 2

The sensor calibration error for Subject 1 and Subject 2 is quantified as helical angle [19] as well and is presented here. Since no video recording is provided for these two subjects, the exact time of each training round is unknown, and therefore the calibration parameters are derived only using the IMU data corresponding to the first detected standing and walking period after the training starts[1] [2].

Table C.1: Sensor calibration error for Subject 1

	Pelvis (°)	Left Thigh (°)	Right Thigh (°)
Stand[10]+Raise Leg[10]	17.91	2.49	1.76
Stand[10]+Walk	16.91	2.60	4.85
Stand+Raise Leg[10]	12.87	6.00	5.16
Stand+walk	11.96	6.17	6.88

Table C.2: Sensor calibration error for Subject 2

	Pelvis (°)	Left Thigh (°)	Right Thigh (°)
Stand[10]+Raise Leg[10]	2.25	3.29	1.58
Stand[10]+Walk	0.24	1.31	7.70
Stand+Raise Leg[10]	8.42	6.08	6.25
Stand+walk	9.06	5.25	9.72

[1]The training starts from 2021-02-05 11:03:14 for Subject 1 in the data series provided by Wilmes[10].
[2]The training starts from 2021-02-05 15:11:39 for Subject 2 in the data series provided by Wilmes[10].

D

Matlab Code for Sensor Calibration

The Matlab code for the proposed automatic sensor calibration algorithm is presented here. The complete code base for this thesis can be found at: *https://github.com/qliu/repository*.

```
% Matlab version: 2021a

%% Initialization

% load raw IMU data
load('IMU_S03_1st_drill.mat');

% Define sliding window parameters for MAD calculations
window_size = 500;
overlap = 25;

% Initialize IMU data container for calibration
fs = 500;
static_duration = 3;
heading_duration = 3;
acc_StaticCalibPelvis = zeros(static_duration*fs,3);
acc_StaticCalibLeft = zeros(static_duration*fs,3);
acc_StaticCalibRight = zeros(static_duration*fs,3);
mag_StaticCalibPelvis = zeros(static_duration*fs,3);
mag_StaticCalibLeft = zeros(static_duration*fs,3);
mag_StaticCalibRight = zeros(static_duration*fs,3);
gyro_HeadCalibLeft = zeros(heading_duration*fs,3);
gyro_HeadCalibRight = zeros(heading_duration*fs,3);

% Parameters controlling the while loop
i = 1;
static_time = 0;
func_time = 0 ;
calib_status = 0;
eps=1e-8;

%% Read in the raw IMU data using the sliding window
while calib_status == 0
    acc_window = acc_S03((i-1)*overlap+1:(i-1)*overlap+window_size,:);
    gyro_window = gyro_S03((i-1)*overlap+1:(i-1)*overlap+window_size,:);
    mag_window = mag_S03((i-1)*overlap+1:(i-1)*overlap+window_size,:);

    % Calculate the MAD value of the raw pelvis acc data in the current window
    mad = MAD(acc_window(:,1:3));

        % If it is identified as 'standing'
        if mad <= 1.5 && static_time-3<-eps
            % Store the corresponding acc and mag data
            acc_StaticCalibPelvis(static_time*fs+1:static_time*fs+overlap,:) = acc_window(1:
                overlap,1:3);
            acc_StaticCalibLeft(static_time*fs+1:static_time*fs+overlap,:) = acc_window(1:
                overlap,4:6);
```

```
            acc_StaticCalibRight(static_time*fs+1:static_time*fs+overlap,:) = acc_window(1:
                overlap,7:9);
            mag_StaticCalibPelvis(static_time*fs+1:static_time*fs+overlap,:) = mag_window(1:
                overlap,1:3);
            mag_StaticCalibLeft(static_time*fs+1:static_time*fs+overlap,:) = mag_window(1:
                overlap,4:6);
            mag_StaticCalibRight(static_time*fs+1:static_time*fs+overlap,:) = mag_window(1:
                overlap,7:9);
            static_time = static_time + 0.05;

        % If it is identified as 'walking'
        elseif mad >= 1.75 && mad <= 3.5 && func_time-3<-eps
            % Store the corresponding gyro data
            gyro_HeadCalibLeft(func_time*fs+1:func_time*fs+overlap,:) = gyro_window(1:overlap
                ,4:6);
            gyro_HeadCalibRight(func_time*fs+1:func_time*fs+overlap,:) = gyro_window(1:
                overlap,7:9);
            func_time = func_time + 0.05;
        end

        % If enough IMU data is collected, end the loop
        if (static_time-3) > -eps && (func_time-3) > -eps
            calib_status = 1;
        end
    i = i+1;
end

%% Vertical calibration

% Pelvis sensor
y_pelvis = mean(acc_StaticCalibPelvis);
y_pelvis = y_pelvis/norm(y_pelvis);
z_pelvis = cross([1 0 0],y_pelvis);
z_pelvis = z_pelvis/norm(z_pelvis);
x_pelvis = cross(y_pelvis,z_pelvis);
x_pelvis = x_pelvis/norm(x_pelvis);
Rvfsf_pelvis = [x_pelvis;y_pelvis;z_pelvis];

% Left thigh sensor
y_left = mean(acc_StaticCalibLeft);
y_left = y_left/norm(y_left);
z_left = cross([1 0 0],y_left);
z_left = z_left/norm(z_left);
x_left = cross(y_left,z_left);
x_left = x_left/norm(x_left);
Rvfsf_left = [x_left;y_left;z_left];

% Right thigh sensor
y_right = mean(acc_StaticCalibRight);
y_right = y_right/norm(y_right);
z_right = cross([1 0 0],y_right);
z_right = z_right/norm(z_right);
x_right = cross(y_right,z_right);
x_right = x_right/norm(x_right);
Rvfsf_right = [x_right;y_right;z_right];

%% Heading calibration

% Left thigh sensor
gyro_vf_Left = gyro_HeadCalibLeft * Rvfsf_left';
gyro_Left = gyro_vf_Left(:,[3,1]);
C_left = pca(gyro_Left,'Centered',false);
gamma_left = -atan2(C_left(2,1),C_left(1,1));
Rafvf_left=[cos(gamma_left) 0 sin(gamma_left); 0 1 0; -sin(gamma_left) 0 cos(gamma_left)];

% Right thigh sensor
gyro_vf_Right = gyro_HeadCalibRight * Rvfsf_right';
gyro_Right = gyro_vf_Right(:,[3,1]);
C_right = pca(gyro_Right,'Centered',false);
gamma_right = -pi-atan2(C_right(2,1),C_right(1,1));
```

```
Rafvf_right=[cos(gamma_right) 0 sin(gamma_right); 0 1 0; -sin(gamma_right) 0 cos(gamma_right)
    ];

% Final transformation matrix for two thigh sensors
Rafsf_left = Rafvf_left * Rvfsf_left;
Rafsf_right = Rafvf_right * Rvfsf_right;

% Using mag data during the standing period and two thigh calibration results to calibrate
    the pelvis sensor
mag_left = mag_StaticCalibLeft*Rafsf_left';
mag_right = mag_StaticCalibRight*Rafsf_right';
mag_pelvis = mag_StaticCalibPelvis*Rvfsf_pelvis';

theta1 = atan2(mean(mag_left(:,1)),mean(mag_left(:,3)));
theta2 = atan2(mean(mag_right(:,1)),mean(mag_right(:,3)));
theta3 = atan2(mean(mag_pelvis(:,1)),mean(mag_pelvis(:,3)));
gamma_pelvis = mean([theta1,theta2])-theta3;

% Final transformation matrix for the pelvis sensor
Rafvf_pelvis=[cos(gamma_pelvis) 0 sin(gamma_pelvis); 0 1 0; -sin(gamma_pelvis) 0 cos(
    gamma_pelvis)];
Rafsf_pelvis = Rafvf_pelvis * Rvfsf_pelvis;


%% Function calculating the MAD value
function mad = MAD(acc)
    temp = sqrt(acc(:,1).^2 + acc(:,2).^2 + acc(:,3).^2);
    mad = sum(abs(temp-mean(temp)))/500;
end
```